Passive Geolocation of Low-Power Emitters in Urban Environments Using TDOA

THESIS

Myrna B. Montminy, Captain, USAF

AFIT/GE/ENG/07-16

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GE/ENG/07-16

Passive Geolocation of Low-Power Emitters in Urban Environments Using TDOA

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Myrna B. Montminy, B.S.E.E.

Captain, USAF

March 2007

AFIT/GE/ENG/07-16

Passive Geolocation of Low-Power Emitters in Urban Environments Using TDOA

Myrna B. Montminy, B.S.E.E.
Captain, USAF

Approved:

Robert J Mills	26 Feb 2007
Dr. Robert F. Mills (Chairman)	date
Richard K. Martin	02/26/2007
Dr. Richard K. Martin (Member)	date
	26 Feb 07
Dr. Michael A. Temple (Member)	date

Abstract

Low-power devices such as key fobs, cell phones, and wireless routers are commonly used to control Improvised Explosive Devices (IEDs) and as communications nodes for command and control. Quickly locating the source of these signals is difficult, especially in an urban environment where buildings and towers can cause interference. This research presents a geolocation system that combines the attributes of several proven geolocation and error mitigation methods to locate an emitter of interest in an urban environment. The proposed geolocation system uses a Time Difference of Arrival (TDOA) technique to estimate the location of the emitter of interest. Using multiple sensors at known locations, TDOA estimates are obtained by cross-correlating the signal received at all the sensors. A Weighted Least Squares (WLS) solution is used to estimate the emitter's location. If the variance of this location estimate is too high, a sensor is detected and identified as having a Non-Line of Sight (NLOS) path from the emitter. This poorly located sensor is then removed from the geolocation system and a new position estimate is calculated with the remaining sensor TDOA information. The performance of the TDOA system is assessed through modeling and simulations. Test results confirm the feasibility of identifying a NLOS sensor, thereby improving the geolocation system's accuracy in an urban environment.

Acknowledgements

I would like to thank my husband. Your support and understanding got me through many hard and long days. Thank you for your love and encouragement.

Myrna B. Montminy

Table of Contents

Page

List of Figures

List of Tables

Passive Geolocation of Low-Power Emitters in Urban Environments Using TDOA

I. Introduction

1.1 Background

Low-power emitters are greatly contributing to the complexity of the electronic warfare problem. They are commonly used to control IEDs, often detonating the IEDs remotely with no warning to those who are targeted. Various emitters can also be used by terrorists for command and control devices. Quickly locating the source of the detonation signal is difficult, especially in an urban environment where buildings and other signals can cause interference. This research presents a geolocation system that use time difference of arrival (TDOA) to locate emitters of interest in an urban environment.

Signals of interest, including signals from airplanes and radars, are detected with receivers located and controlled by operators at a safe distance. Now, emitters of interest include key fobs, cordless phones, cell phones, wireless routers, and walkie-talkies. Together these devices operate across a wide range of the radio frequency spectrum. The diversity of these emitters presents a significant force protection challenge because little is currently known about how to locate them.

The urban environment introduces additional challenges, including multipath, signal scattering, and widespread interfering signals. These complications make it difficult and sometimes impossible to detect emitters, especially from the safe stand-off distance that most geolocation systems currently use. New systems must have the ability to approach much closer to the emitter to detect its signal, while keeping the military controller safe from harm. This requirement means its possible deployment onto an unmanned aerial vehicle (UAV) or unmanned ground vehicle (UGV). Deploying the geolocation system onto a UAV adds size and weight restrictions to the

already complex problem. These challenges highlight some of the many aspects that this difficult problem presents.

1.2 Problem Statement

The purpose of this research is to provide a background on geolocation methods and recommend methods that can be used to locate emitters within an urban environment through modeling and simulation. This capability can lead to alternative approaches to exploit the enemy, not only by locating possible IED detonators, but also locating cell phones and communications nodes that may be used by the enemy for command and control purposes.

1.3 Current Research

When a signal arrives at two moving, spatially separated receivers, the receivers measure a difference in phase and frequency. The time that a signal arrived at two spatially separate receivers also yields helpful information. These characteristics are the basis for four methods of geolocation: 1) the angle of arrival (AOA) method that locates a position using the directional angle of a signal, 2) the frequency difference of arrival (FDOA) that determines the position of the emitter from the difference in frequency of the signal measured between two receivers, 3) the time of arrival (TOA) technique that calculates the position of the emitter using the precise time the signal arrives at multiple receivers, and 4) time difference of arrival (TDOA) which uses the difference in time a signal is received at two or more receivers to determine the location of an emitter. In passively locating emitters, the TDOA and FDOA approaches offer several advantages over both the AOA and TOA methods. Current research of each of these emitter location techniques is discussed in more detail in Chapter II.

1.4 Scope and Application

This research investigates a process for locating low-power devices. As mentioned earlier, the emitters of interest are found in a wide variety of applications

ranging from remote controlled toys, to household electronics, to wireless communications. The research presented provides background on several current geolocation methods and modify and apply them to locate emitters in the presence of multipath and noise that are abundant in urban environments.

1.5 Assumptions

Specific limits and assumptions are imposed to make the problem solvable within time and equipment availability constraints. It is assumed that the emitter of interest is stationary and within detection range of several sensors. While the emitter location is unknown, the emitter's signal structure, including general waveform and bandwidth, can be recognized. Approximation methods for estimating a single solution from a set of multiple solutions are chosen from those readily available to the technical community, including the estimation work of Knapp and Torrieri [15, 28]. These methods are explained in more detail in Chapter II. It is also assumed that there are no other signals in the area that may cause interference or overlap with the low-power emitter's signal being located.

1.6 Research Benefits

Results from this research are anticipated to have several applications. First, the results will lead the U.S. military one step closer to having the ability to passively locate unknown emitters. For this research, passive geolocation is defined as the ability to geolocate an emitter by processing its radio frequency (RF) emissions. This allows the emitter to be geolocated without being interrogated. Passive capability is crucial in locating emitters that may be used for IED detonation, hostile jamming, or as communications nodes because it can be accomplished without alerting the enemy. It may also benefit search and rescue missions and law enforcement surveillance. Second, the algorithms and geolocation methods that are developed as a part of this research can be applied to current systems or added later to mobile receiver systems. Finally,

a complete geolocation system will be proposed that incorporates several different research efforts in this area.

1.7 Standards

For this research to be successful, the location of an emitter must be accurate to within ten meters. This accuracy is important in urban environments because other emitters may be near the emitter of interest. If the error is greater than ten meters, other emitters within the error range may be mistaken for the emitter of interest. Ten meters is a reasonable distance because it can narrow down the location of the emitter of interest to a specific building, park, or section of street. Accurate geolocation is essential in the military's ability to weaken the enemy and protect its troops.

1.8 Approach / Methodology

This research begins by investigating current methods of locating signals to determine the best approach to locate an unknown emitter. This preliminary investigation and its results can be found in Chapter II. After deciding on a geolocation approach, factors that may impede the goals of this research are considered. These factors include emitter signal strength, tolerable noise level, and estimation schemes. As these factors become more defined, more assumptions may be needed.

The research then focuses on the main problem: locating an unknown emitter in an urban environment. As mentioned previously, the urban environment introduces many obstacles that make it difficult to detect and locate signal sources. Buildings can reflect signals, allowing multiple copies of the same signal to arrive at a receiver at different times and signal strengths, also known as multipath. Many other emitters can produce signals spanning a wide range of the frequency spectrum. These added signals can cause interference and may be strong enough to hide the emitter that needs to be located. The TDOA method of signal geolocation has been widely used to locate signal sources. This research combines the TDOA geolocation method with error detection techniques to effectively locate an emitter of interest. The impact the

urban environment has on detecting and locating emitters will be determined. New methods and algorithms will be developed to cope with these new obstacles.

A new method of signal geolocation is developed and simulated that can detect and identify receivers having only a NLOS signal path to the emitter of interest. These sensors, identified as NLOS, are removed from the TDOA correlation measurements, and new location estimates are calculated. Moving the NLOS receivers to new locations in an attempt to achieve LOS is also explored.

1.9 Equipment Needed

Material needed for this research effort was minimal. All signals and receiver parameters used in this research are simulated using Matlab® Version 7.0 developed by Mathworks, Inc. Simulations are run on a 3.60 GHz Xeon PC. Matlab® is also used extensively to simulate the near-real world geolocation algorithms as they are developed. Algorithms from previous research are used as a basis [17,18,23]. This research effort develops algorithms to simulate correlation and estimation of the location of an emitter in an urban environment.

1.10 Thesis Organization

Background information, including an overview of methods and techniques needed in a geolocation system, is provided in Chapter II. Chapter III details the simulation methodology, discussing the test signals used as well as the methods combined to accurately estimate an emitter's location in an urban environment. Analysis of the results is discussed in Chapter IV. Chapter V provides conclusions from the research effort and recommendations for future research.

II. Background

2.1 Overview

This chapter reviews literature of several geolocation methods and estimation techniques that may be used to passively locate low-power emitters while identifying and mitigating possible sources for error.

2.2 Methods for Locating Emitters

A wide range of methods are currently being used to geolocate signals. The four fundamental techniques include: 1) the AOA method that locates a position using the directional angle of a signal, 2) the FDOA method that determines the position of the emitter from the difference in frequency of the signal measured between two receivers, 3) the TOA technique that calculates the position of the emitter using the precise time the signal arrives at multiple receivers, and 4) TDOA which uses the difference in time a signal is received at two or more receivers to determine the location of an emitter. In passively locating low-power emitters, the TDOA approach offers several advantages over both the AOA and TOA methods. These emitter location techniques are now discussed in more detail.

2.2.1 Angle of Arrival (AOA). The emitter location technique of using the AOA of a signal received at multiple receivers at known locations is regularly used in surveying, radar tracking and vehicle navigation systems [22, 27]. An AOA estimate is made by electronically steering an adaptive phased array antenna in the direction of the arriving emitter's signal. An adaptive phased array antenna is made up of an array of sensors and a real-time adaptive signal processor. A line of bearing (LOB) is calculated from each AOA estimate and drawn from its corresponding receiver location. These LOBs intersect at the estimated location of the emitter [20]. This method is commonly known as triangulation. An illustration of triangulation is shown in Figure 2.1, in which three fixed receivers (A, B, C) provide LOBs that are used to estimate the location of an emitter (P).

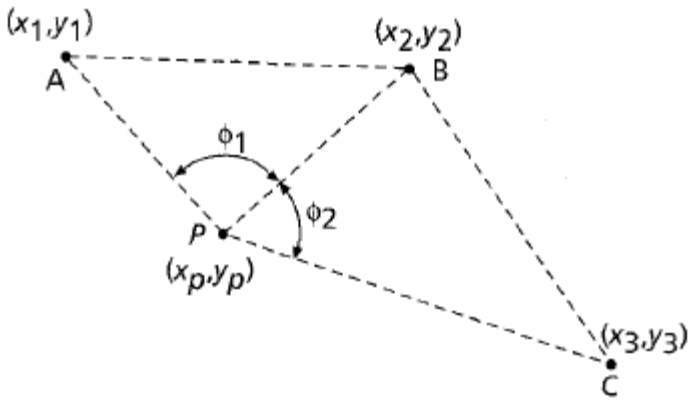


Figure 2.1: Triangulation using three receiver LOBs [22].

The AOA method allows an emitter's location to be determined with just two receivers, fewer than what is necessary in the TDOA method discussed later. Also, no time synchronization between the receivers is required [16, 27]. Both of these advantages reduce the number of receivers needed to measure the AOA of an incoming signal, but extra processor time is needed to calculate and maintain accurate calibration of the antenna arrays. Relatively large and complex hardware is also used [16]. This makes AOA a difficult location technique to use on a UAV or other smaller platform. A significant disadvantage in using AOA in an urban environment is its dependence on a direct LOS path from the receiver to the emitter. Urban environments generate considerable reflections that can cause multiple occurrences of the same signal coming from directions not that of the emitter (also known as multipath). These multipath signals can cause excessive errors in the AOA measurements [22].

2.2.2 Frequency Difference of Arrival (FDOA). In the case that the emitter or one of the receivers is moving, a frequency shift will occur, causing each receiver to receive the low-power emitter's signal at a different frequency. The difference in these frequencies of arrival, or FDOA, can also be used to estimate the emitter's position [25]. FDOAs form a set of points that represent possible emitter positions. Figure 2.2 shows the resulting set of points from three example FDOA measurements calculated from two receivers.

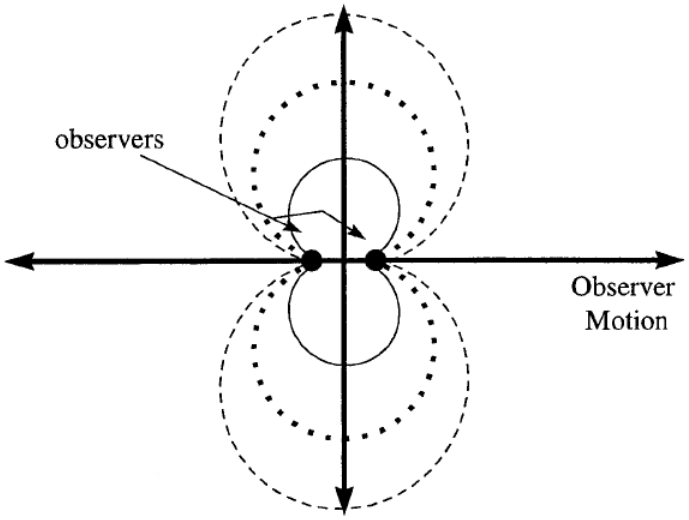


Figure 2.2: FDOA contours [25].

The FDOA method has several limitations in its ability to locate low-power emitters. The relative velocity between the two receivers must be large enough for the FDOA to outweigh the frequency measurement error [14]. The receivers also need to have the ability to measure frequency with more accuracy than the smallest frequency shift that is expected. If this accuracy is not possible, the FDOA may be undetectable below the noise. FDOA is difficult to implement, and very costly, especially when the signal is already at a very low power level [25].

2.2.3 Time of Arrival (TOA). Using the time that a signal arrives at a receiver is another technique that is used in determining the location of an emitter. Because signals propagate at the constant speed of light, the distance from a transmitter to a receiver can be calculated from the propagation time of the signal [22]. This estimated distance forms a circle around the receiver. The intersection of three or more of these circles provides an estimate of the location of the emitter.

For the TOA method to work, specific knowledge of the emitter's signal must be known. TOA uses precise time synchronization of all the receivers, as well as the emitter of interest. The exact time the signal left the emitter must be known in order to determine the signal propagation time [22, 30]. In this research the signal type

and waveform are known, but because the timing is unknown, there is no way to synchronize its time source with each receiver's time source to obtain the precise time when the signal left the emitter.

2.2.4 Time Difference of Arrival (TDOA). The TDOA-based approach to loca-ting emitters is one of the most commonly used position location techniques [30]. TDOA is used in many geolocation, operational and scientific applications [3] and is widely used in sonar and radar to find the position of a signal of interest [31]. Many navigation systems, including long range navigation (LORAN) and Decca, use the TDOA between a radio signal and multiple stations to determine a desired navigation position [10].

The TDOA technique uses the differences in the time that a signal arrives at multiple receivers. Each TDOA measurement produces a hyperbolic curve yielding a set of possible positions. The equation of this hyperboloid is given by

$$R_{i,j} = \sqrt{(X_i - x)^2 + (Y_i - y)^2 + (Z_i - z)^2} - \sqrt{(X_j - x)^2 + (Y_j - y)^2 + (Z_j - z)^2} \tag{2.1}$$

where the three-dimensional points (X_i, Y_i, Z_i) and (X_j, Y_j, Z_j) represent the pair of receivers, i and j. A two-dimensional emitter's location can be estimated from the intersection of two or more hyperbolas generated from three or more TDOA measurements. An emitter's three-dimensional location estimate needs at least four TDOA measurements to calculate its location [22]. A two-dimensional example is shown in Figure 2.3. The hyperbolas in Figure 2.3 are composed of possible solutions generated from the time differences, $\triangle t = t_i - t_1$. From the intersection of the hyperbolas, the location of the emitter, p is estimated [19].

Locating an emitter using the TDOA method is possible even if multipath is present. This method's ability to overcome multipath is a significant advantage over the previous methods, especially in locating emitters in urban environments where buildings and towers can cause significant reflections. For the TDOA technique to be

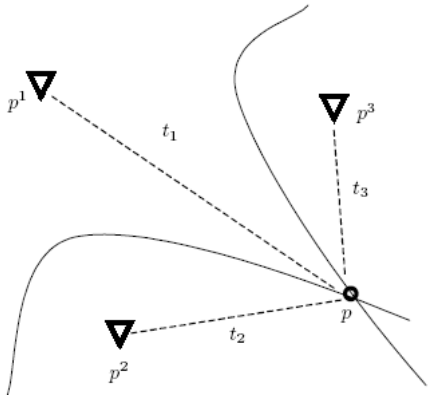


Figure 2.3: TDOA hyperbolic curves [19].

successful, at least one LOS path between the emitter and each receiver must exist. If there is a LOS path from the emitter to a sensor, and multipath is present, the signal with the least time delay (as determined by correlation described later) is the LOS signal [16]. The condition when there is no LOS path between the transmitter and a receiver is discussed more in Section 2.6.2.

Another advantage in using TDOA to passively locate unknown emitters (over the other methods presented) is that there is no requirement that the emitter be synchronized in time with the receivers. When using the TDOA technique to locate the source of a signal, the time that the signal left the emitter is not needed, and only the receivers need to be synchronized in time [22, 30]. A timing reference that is commonly used is global positioning system (GPS), but in urban environments, access to GPS satellites is often obstructed by buildings or towers. Another approach that is used in time synchronization is the use of atomic clocks, like a Rubidium or Cesium time source [16]. The ability to locate a signal using TDOA without needing to time synchronize the emitter is especially important in detecting signals passively. In passive detection, the characteristics of the signal coming from the emitter are unknown, so there is no way to obtain the exact time when the signal left the emitter [15].

Estimation of an unknown emitter's position depends on the ability to measure the time differences of the signal as it arrives at several different receivers. These

time difference measurements can be accomplished using time correlation methods described in the next section.

2.3 Time Difference of Arrival Estimation (TDOA)

Typically two spatially separated receivers will receive the same emitted signal at different times. An example configuration is shown in Figure 2.4. This configuration is represented by received signal equations

$$x(t) = s(t) + n_x(t) \tag{2.2}$$

$$y(t) = \alpha s(t - D) + n_y(t), \tag{2.3}$$

where $s(t)$ is the original signal, $n_x(t)$ and $n_y(t)$ are uncorrelated zero-mean Gaussian noise processes, α is the scaled difference in amplitude between the two received signals, and D, which can be either positive or negative, is the TDOA between the two signals [31].

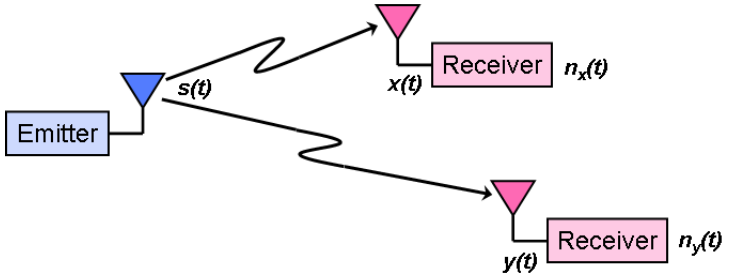


Figure 2.4: Example configuration of two receivers and one emitter.

After the signal is acquired by the two receivers, correlation analysis is performed. This correlation yields the difference of time that each receiver actually received the signal. Several of these time measurements, converted to distance, are used to calculate the location of the low-power emitter of interest, as previously discussed. This section reviews two methods that are used to correlate two received signals to find the TDOA. The cross correlation function can be used when the emitter and the two receivers are stationary. The complex ambiguity function can be used to estimate

both the FDOA and the TDOA for cases when there is motion among the emitter or the receivers.

2.3.1 Cross Correlation Function. The first step in locating an emitter is estimating the TDOA. For this research, the signals are assumed to be real-valued. A commonly used method to obtain this measurement is by using the generalized cross correlation function

$$R_{xy}(\tau) = E[x(t)y(t-\tau)], \tag{2.4}$$

where E is the expected value operator.

Figure 2.5 illustrates the process of the cross correlation function when it is applied to find the TDOA of a signal. Assuming ergodicity, the cross correlation in (3.2) can generally be written as

$$\hat{R}_{xy}(\tau) = \frac{1}{T-\tau}\int_{\tau}^{T} x(t)y(t-\tau)dt, \tag{2.5}$$

where T is the observation interval. The value of τ that maximizes the cross correlation function in (3.3) is the time delay estimate, $\hat{R}$ [15]. A visual example of this is also shown in Figure 2.6.

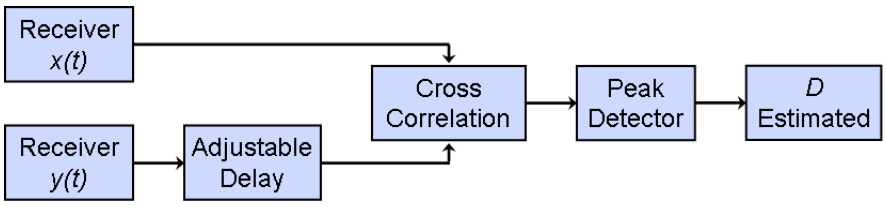


Figure 2.5: Correlation of two received signals, $x(t)$ and $y(t)$ to obtain the TDOA.

Error analysis of the cross correlation function shows that adding a filter to the input of each signal prior to correlation improves the accuracy of the time delay estimate [15]. In the frequency domain, adding a filter is equivalent to multiplying the power spectral density by a weighting function. Knapp and Carter [15] describe five weighting functions that are designed to reduce noise for different cases. Function

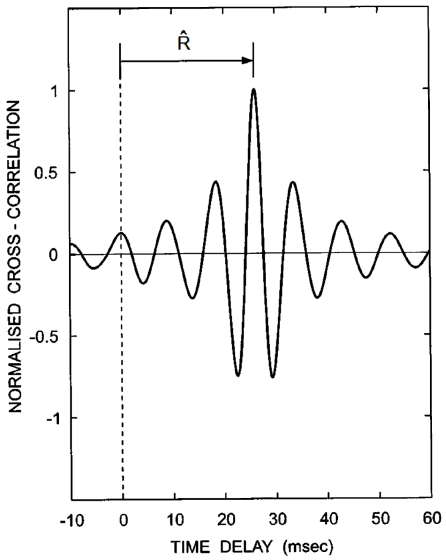


Figure 2.6: Cross correlation example. $\hat{R}$ is the estimated time delay [11].

selection requires some knowledge of the transmitted signal's frequency spectrum and noise [31], which may not be readily available for an unknown signal. But if characteristics like the waveform, or expected signal power are known, using a weighting function may prove to be beneficial in correlating the signal to find its TDOA.

2.3.2 Correlation Window Size. The ability to correlate two signals also depends on the autocorrelation of the original signal from the emitter that is being located. By increasing the correlation window size (T_s), the autocorrelation of a signal shows a more narrow peak. When the autocorrelation of a signal has a more narrow, defined peak, the chance for error and ambiguities to occur when correlating the signal after it is received by two spatially separated receivers is lower than in the case when the autocorrelation has a wider peak [17]. Various GSM autocorrelation functions using different correlation window sizes (T_s) are shown in Figure 2.7. In this figure, T_s is also equal to the total signal length. Increasing the sampling rate may also benefit correlation of two signals. The correlation window size is especially important when the emitter produces a low-power signal, causing the SNR to also be low. This is further developed in Chapter III.

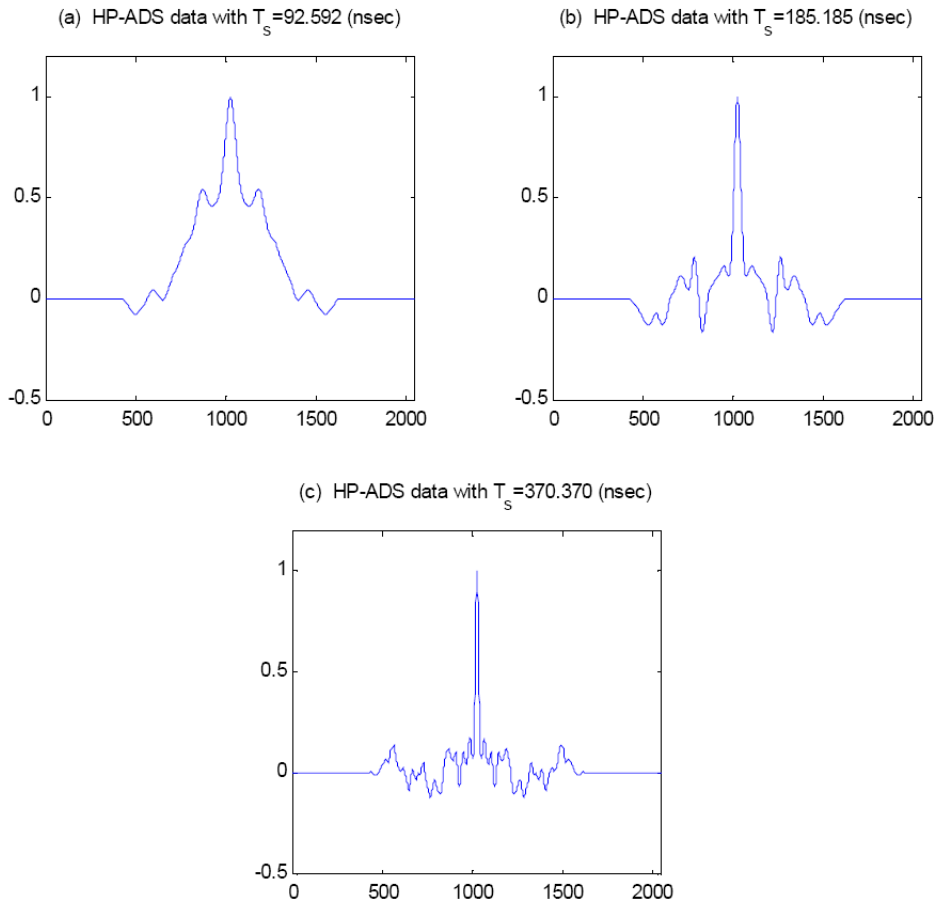


Figure 2.7: Auto-correlations of GSM signal with various correlation window sizes, Ts [17].

The methods discussed to decrease the chance for ambiguities and errors while correlating two received signals will have to be balanced with the time and processor requirements of the geolocation system and its ability to locate an unknown emitter in the time required.

2.3.3 Ambiguity Function. The ambiguity function is viewed as an extension of the generalized cross correlation function for moving receivers or transmitters. It can be used to jointly estimate both the TDOA and FDOA of a signal at two spatially separated receivers [24]. With the addition of phase shift, the received signals become

more complicated, as shown in the equations

$$x(t) = s(t) + n_x(t) \tag{2.6}$$

$$y(t) = A_r s(t - D)e^{j2\pi f_d t} + n_y(t), \tag{2.7}$$

where f_d is the FDOA and D is again the TDOA between the two receivers. The complex ambiguity function is given as [24]

$$|A(\tau, f_d)| = \int_0^T x(\tau)y^*(t - \tau - D)e^{-j2\pi f_d t}e^{+j2\pi f_d \tau}dt, \tag{2.8}$$

where τ corresponds to the estimated TDOA, $\hat{D}$. τ and f_d are simultaneously solved to maximize $|A(\tau, f_d)|$, resulting in the estimated TDOA and FDOA of the signal between two receivers [24].

Using both the TDOA and FDOA methods allows the estimate of the emitter's location in both poor noise conditions as well as in the presence of many spectrally and spatially overlapping interfering signals. Figure 2.8 shows a typical complex ambiguity function in which TDOA and FDOA can be determined [26].

Both the generalized cross correlation function and the ambiguity function suffer in poor SNR environments. Thus, if implemented techniques discussed previously to reduce noise at the receivers (filtering) may have to be considered [26], especially for a low-power emitter.

2.4 Estimation of Low-Power Emitter Location

The previous section described how correlation can be used to estimate the TDOA between two received signals. Once the TDOA between all pairs of receivers has been estimated, the location of the low-power emitter can be solved by finding the intersections of the formed hyperbolas with respect to the known receiver locations, as discussed earlier. But because the equations of the hyperbolas are nonlinear, estimating the solution through brute force calculations can be time consuming and

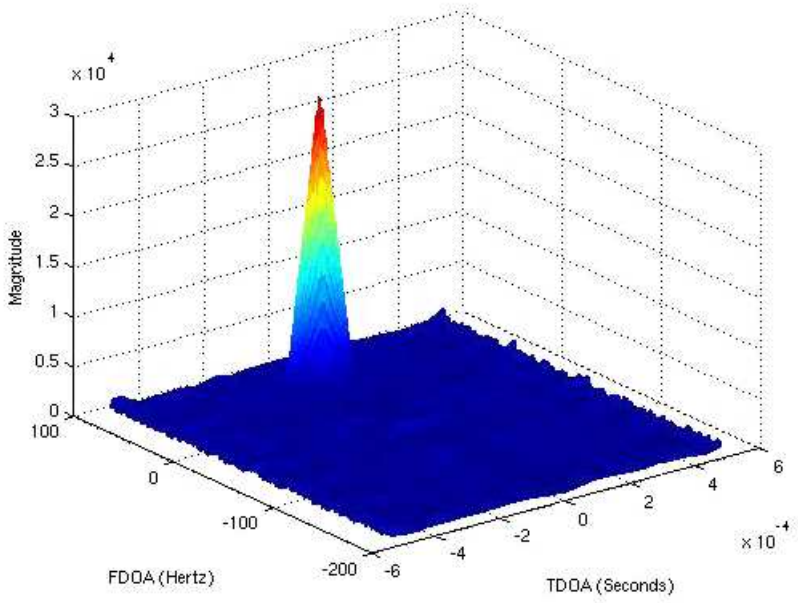


Figure 2.8: Plot of example complex ambiguity function output [13].

computationally expensive, especially when noise and error are considered. Complexity also increases when the receivers are distributed arbitrarily. This causes a non-linear relationship between the TDOA measurements and the position of the emitter to develop, resulting in a non-convex maximum likelihood (ML) function that is difficult to solve. Various estimators have been developed that solve this problem through an iterative solution process [29].

Locating low-power emitters must be completed in a short amount of time, optimally as close to real time as possible, to ensure that the emitter of interest does not have a chance to move before its location is estimated and there has been time to act on that information. For this research, the time the geolocation system needs to estimate the position of the emitter is referred to as the processing time. A balance needs to occur between the processing time needed and the resulting error in the estimated location for the geolocation system to be successful. Several approaches have been proposed to solve the ML function, time constraints limit the methods

that can be used to estimate the emitter's location. Those methods that may cut down processor time are discussed in this section.

2.4.1 Taylor Series Approximation. One of the simplest methods to find a solution of hyperbolic TDOA equations is to use a Taylor-series expansion to linearize the hyperbolic equations. Although commonly used, this method needs an initial guess close to the true solution to avoid being stuck in a local minima. It also improves the location estimate in a series of steps, determining the local linearization solution (LS) at each step. There is no guarantee of convergence to a solution, especially when the initial guess is not close to the true solution [4, 12]. Because little is known about the emitter of interest, it would be difficult to guess the initial location required to estimate the position, especially if this were implemented on an automated system without a human decision maker. Also, calculating the LS during each estimation step adds to the calculations that are needed and processor time. This added time may limit the geolocation system's ability to locate an emitter in near-real time.

2.4.2 Closed-Form Solution. Another method of estimating the location of an emitter from TDOA measurements is a closed-form technique. In comparison to other emitter location estimation methods, closed-form methods may not be as accurate. Chan and Ho [4] have developed a closed-form solution that can be used to determine an emitter's location for both near and far sources. This method only works when the signal has a large SNR, and the number of receivers must be four. Fang [10] developed a method that estimates a unique position for the case when the number of TDOA measurements is equal to the number of unknown coordinates to the emitter. Most closed-form solutions, although they require less calculations than the Taylor-series expansion method, cannot take advantage of extra receivers that may improve solution accuracy [4].

2.4.3 Hyperbolic Asymptotes. An additional method of localization using TDOA measurements has been developed that is less computationally intensive, but

may also be less accurate than the previous estimation methods referenced. Drake and Doğançay [9], proposed a new, simplified method to locate an emitter by using the set of linear equations that corresponds to the hyperbolic asymptotes of the TDOA measurements.

Figures 2.9 and 2.10 show hyperbolas and their respective hyperbolic asymptotes [9]. This method of estimating an emitter's location reduces the calculations needed considerably. However, it also introduces more room for error, especially if the receivers are relatively close to the emitter, or to each other.

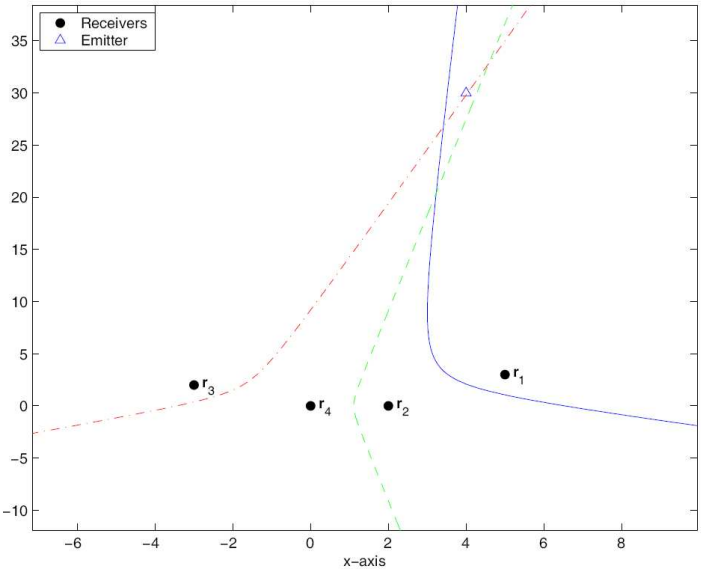


Figure 2.9: Geolocation using TDOA hyperbolas [9].

2.5 Over-Approximation

Because LOS signal paths to the emitter are more difficult to obtain in an urban environment, more receivers than are minimally required to estimate a position must be deployed, so more TDOA measurements may be obtained than are necessary to estimate the emitter's position. But some receivers may be further from the emitter than others, or there may not be a direct path from a receiver to the emitter, which could add more uncertainty to the estimated location. There are a few estimation

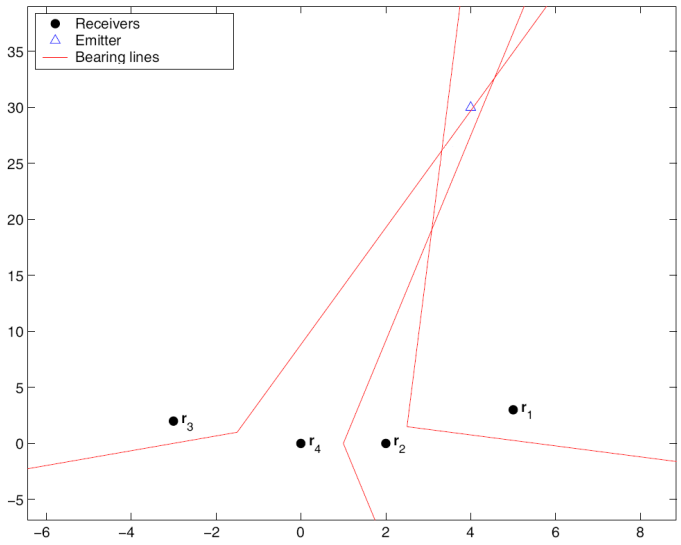


Figure 2.10: Geolocation using hyperbolic asymptotes [9].

methods that take into account extra TDOA measurements. One of these methods is the Divide and Conquer (DAC) approach.

2.5.1 Divide and Conquer Position Estimation Method. The DAC estimation method divides the TDOA measurements into equal groups each containing the same number of unknowns. The unknown parameters are calculated in each group, then combined to give the final position estimate [2]. This solution uses stochastic approximation and requires a large amount of data, which limits the amount of noise that can be tolerated [4]. This is a significant disadvantage in estimating the location of a low-power signal, where the noise level may be relatively high compared to the signal power. This method also assumes that the receivers have a good LOS signal path to the emitter, and all TDOA measurements are good. It does not distinguish between measurements taken from a receiver close to the emitter and one that may be further away that may cause more error in its relative TOA measurements.

2.5.2 Linear Weighted Least Squares (WLS) Algorithm. In real environments, TDOA measurements contain noise, either from inconsistencies in the receivers, or from other outside sources. Torrieri [28] derived the principal algorithms

in 1984 that are still commonly used to estimate an emitter's position from TDOA measurements [29]. Torrieri's weighted least squares (WLS) estimator, when applied to finding the location of an emitter using TDOA measurements, can successfully compute a location estimate from any number of sensors and their respective TDOA measurements.

The location of an emitter is estimated from TDOA measurements using an iterative linear WLS approach. WLS is an estimation method that is similar to the more common LS method. The WLS estimation method uses the same minimization of the sum of the residuals:

$$S = \sum_{i=1}^{n} [y_i - f(x_i)]^2, \tag{2.9}$$

where (x_i, y_i) are the data points for $i = 1, 2, ..., n$ cases. For position estimation using TDOA, the TDOA values and sensor location information serve as the input data points. The function f that produces $f(x_i) \approx y_i + \epsilon_i$ gives the solution, an estimated emitter position.

The WLS method of estimation is similar to the LS approach, but instead of weighting all the points equally as in (2.9), the points are weighted such that the points with a greater weight contribute more to the fit:

$$S = \sum_{i=1}^{n} w_i [y_i - f(x_i)]^2. \tag{2.10}$$

The weight, w_i, is usually given as the inverse of the variance, giving points with a lower variance a greater statistical weight [28]:

$$w_i = 1/\sigma_i^2. \tag{2.11}$$

Applying this WLS method may cut down on the number of iterations needed for the solution to converge because there will be less variance in each estimate, further reducing the overall time the processor needs to make all the calculations. But as the

number of sensors and their respective TDOA measurements increase, the number of calculations needed will rise, increasing the processing time.

As the methods presented in this section suggest, over-approximation still remains a problem in locating the source of a signal. A new estimation method that takes advantage of additional TDOA measurements than are necessary is developed in Chapter III.

2.6 Urban Environment Characteristics

In a dense urban environment, there may not always be direct LOS from a receiver to the emitter of interest. Buildings and towers can cause the signal to be blocked or reflected, causing multiple paths to occur. Figure 2.11 shows an example of several receivers and the respective signal paths to an emitter of interest.

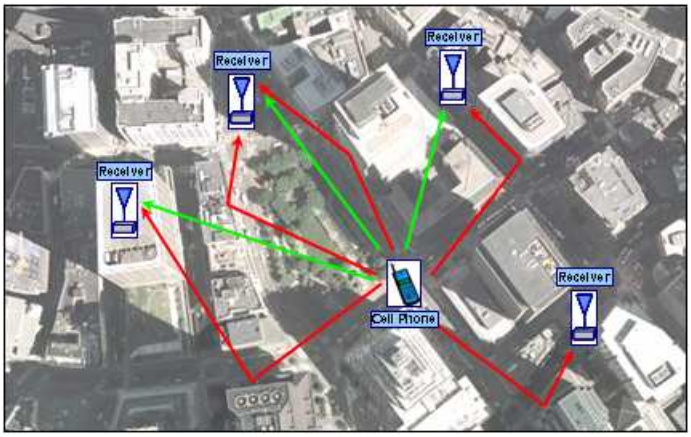


Figure 2.11: Example layout of receivers and emitter in an urban environment with possible signal propagation paths displayed. City background from [1].

2.6.1 Multipath. As discussed earlier, assuming that a direct LOS exists between the receiver and emitter, when multiple delayed copies of the signal arrive at a specific receiver, also known as multipath, the signal with the least time delay is the LOS signal [16]. The LOS path is assumed to be the most direct path to the emitter [22]. Figure 2.12 shows an example of how the same signal can be received multiple times at one receiver.

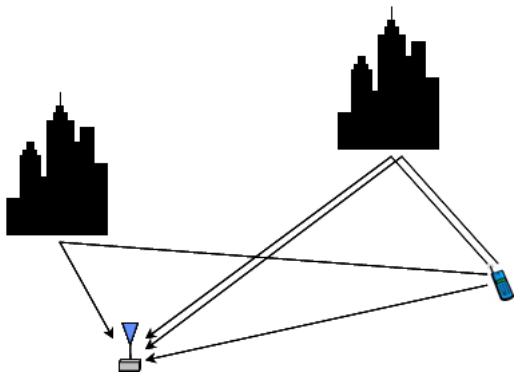


Figure 2.12: Example of multipath signals.

2.6.2 Non-Line of Sight (NLOS). In locating an emitter, position error and complexity is added if an obstacle like a building or tower blocks the LOS path between the sensor and emitter. In this case, the most direct signal path from the emitter to the sensor is not the LOS path, leading to error and variance in the estimated position of the emitter. Non-LOS (NLOS) paths from the emitter to the sensors are common, especially in the urban environment. There is a great need to have the ability to determine which sensor in a geolocation system is limited to a NLOS path from the emitter. If the NLOS sensor can be identified and removed from the system, the error associated with this sensor's TDOA calculations can also be removed, leading to a more accurate emitter position estimate [7]. Figure 2.13 shows an example configuration that includes a NLOS receiver.

Identifying a NLOS sensor is crucial in reducing the error in estimating the position of a low-power emitter. If a sensor has a NLOS propagation path to the emitter, the signal has to travel a longer distance to reach the sensor. This causes the estimated position to be further away from the NLOS sensor and away from the true position of the emitter. Figure 2.14 shows the effect a NLOS receiver can have on the overall TDOA position estimate. In this configuration, the mobile station (MS) is the emitter, and each base station (BS), is a sensor.

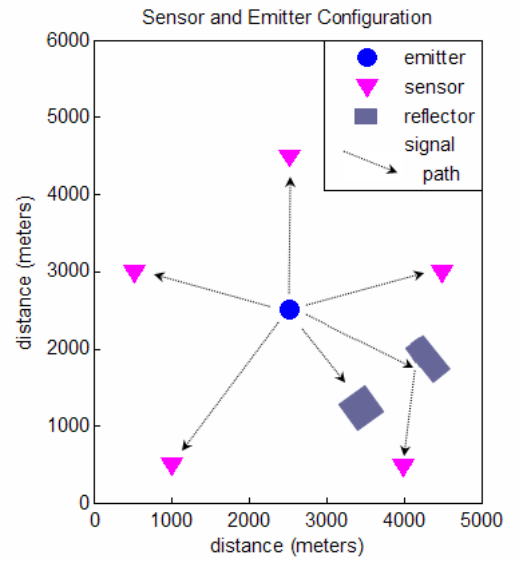


Figure 2.13: Example of NLOS sensor and signal path.

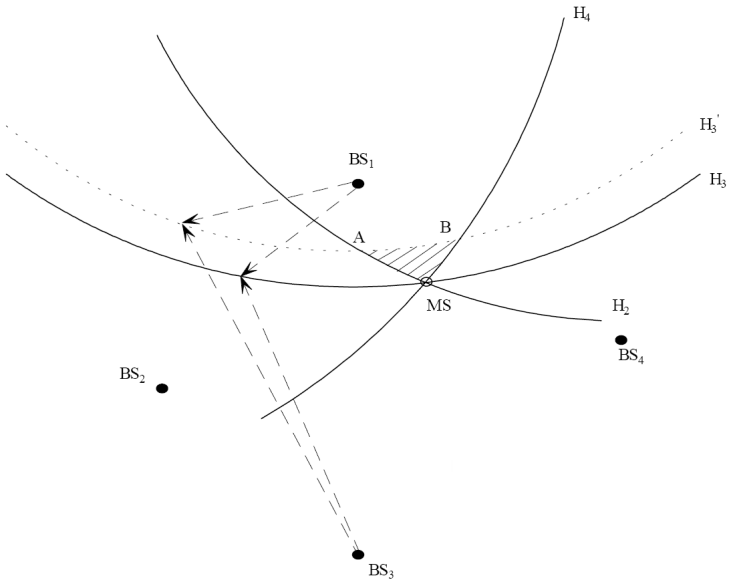


Figure 2.14: NLOS effect on TDOA location estimate. The positions A and B are the new estimates as a result of the NLOS sensor, BS3 [6].

2.7 Receiver Error Identification and Mitigation

NLOS error mitigation techniques have been researched extensively, but most techniques assume that measurements with NLOS errors only consist of a small portion of the total measurements and can be treated as outliers. If there are multiple

NLOS sensors present, these measurements make up a larger portion of the total measurements, and can have a much larger effect on the final estimated position [7].

Several error mitigation methods using deletion diagnostics have been developed to work around the problem of multiple NLOS receivers. The time-history based approach and the method of ranking the residual of each receiver have both been adopted by many to reduce the effect a NLOS receiver may have on the overall emitter position estimate. There is also research into how the geometrical layout of the sensors affects the estimated position of the emitter of interest. This section describes these methods.

2.7.1 Time-History Based Approach. Wylie and Holtzman [32] developed a time-history based hypothesis test that identifies the NLOS error in TDOA calculations by looking at each base station's relative TOA measurement over a period of time, then combining this information with the variance of the standard noise measurement. Their work was focused on cell phone systems.

This NLOS identification technique then compares the standard deviation of a sample statistic (each base station's relative TOAs and the known standard noise for each receiver) to the known standard deviation of that statistic when the measurements are from a LOS receiver. When there is a LOS from the base station to the MS, then the standard measurement noise's effect on the measured TOA can be predicted. If there is NLOS error present, then the measured TOA is expected to have a significantly larger average deviation from the expected curve than when there is no NLOS error present [32]. This method works well to detect if a receiver has only a NLOS signal path to the emitter, but cannot differentiate which receiver has the NLOS signal path.

2.7.2 Residual Ranking. One general method adopted by many researchers [5, 32] uses a residual rank test. In this method, all of the initial relative TOA measurements, t_i, are used to estimate the emitter's position. A NLOS error in a receiver

can cause inconsistent TOA measurements, leading to inconsistencies in the TDOA calculations as well. After several emitter position estimates are completed over time, the NLOS error will become more detectable due to its irregularities.

A residual rank test can be performed to determine which receiver is most likely to have the NLOS signal path to the emitter. The general steps are:

1. Use all TDOA measurements at t_i to estimate the position.
2. Calculate each receiver's residual:

$$e_m(t_i) = r_m(t_i) - \hat{L}_m(t_i), \tag{2.12}$$

   where $r_m(t_i)$ represents the measured TDOAs, and $\hat{L}_m(t_i) = |\hat{d}(t_i) - d_m|$ are the calculated TDOAs.
3. Count number of times $|e_m(t_i)|$ is largest error for each t_i.
4. Rank results from 3.

The receiver with the consistently largest residual, $e_m(t_i)$, is assumed to be the receiver with the NLOS signal path to the emitter [32]. This method is effective in determining if there is a NLOS receiver, but needs several position estimates over an extended period of time to allow the NLOS receiver's relative TOA measurements have a detectable effect on the overall position estimate of the emitter. While effective, this method may take too much time to allow for near-real time emitter position estimates. This also does not account for the case of more than one NLOS receiver.

2.7.3 Receiver Geometrical Layout. Because of the recent government push towards E-911, most research into passively locating emitters assumes a typical cell phone BS geometrical configuration. In this layout, the cell phone emitter is surrounded by BS's that are evenly spaced azimuthly around the cell phone [7]. An example of this is in Figure 2.15.

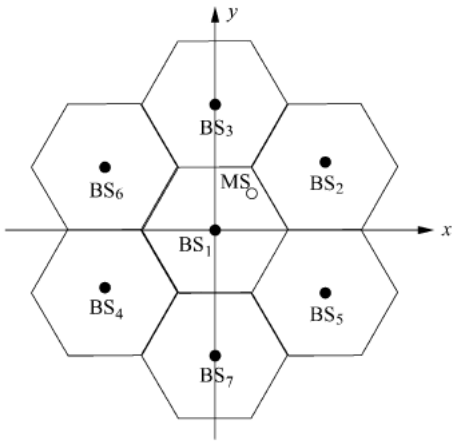
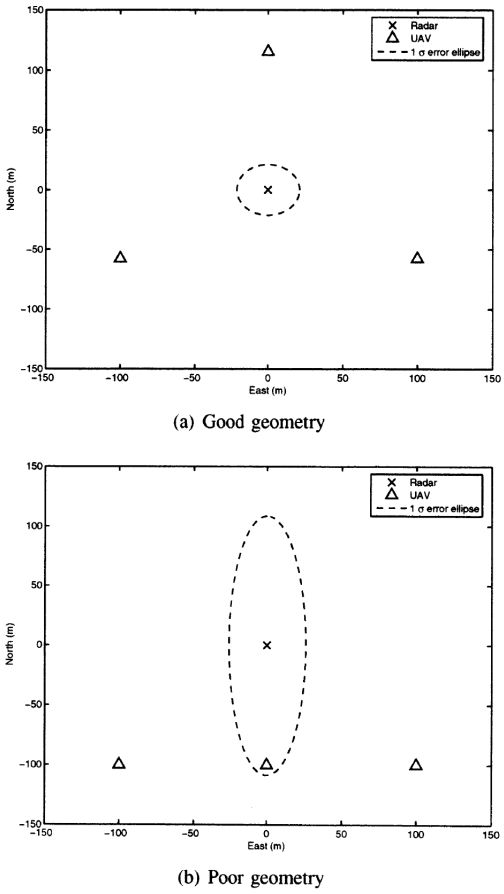


Figure 2.15: BS typical layout [7].

More recently, Drake [8] explored receiver geometrical layouts as they would happen if mounted on multiple UAVs. Using statistical analysis developed by Torrieri [28], Drake was able to accurately predict error bounds for various receiver layouts. These are shown in Figures 2.16a and 2.16b.

If the receivers are mounted on UAVs, the ability exists to move the receiver that is causing the error to help mitigate the multipath NLOS error. If a sensor is identified as located in a poor location with respect to the other sensors, causing more error in the estimated position of the emitter, that sensor can be moved to a better location. Based on the error predictions of various sensor configurations, better sensor geometrical layouts can be recommended and implemented while the UAVs are in flight, assuming that an initial position estimate can be obtained.

Techniques of identifying a NLOS receiver work well when there is only one receiver with a NLOS path from the emitter and the sensors are located surrounding the emitter [7]. But many times more than one BS suffers from NLOS, or more than one receiver is located too far away from the emitter to help in estimating the emitter's location. The case when more than one receiver contributes to bad TDOA measurements is further developed in Chapter III.



(a) Good geometry

(b) Poor geometry

Figure 2.16: TDOA location uncertainty elipses for good (a) and poor (b) geometry [8].

2.8 Summary

This chapter provides a foundation on geolocation systems, reviewing several methods that can be used to locate signal sources, specifically, those methods that can be applied to passively locate low-power emitters. The TDOA emitter locating technique is the most appropriate method to find the source of a low-power signal. It is less computationally intensive and physically less complex than the AOA and TOA methods. For these reasons, TDOA was the method of choice for this research.

There are several major challenges in developing methods to passively locate low-power emitters. Size, weight, power and cost all have important roles and are considered in the design of the hardware and algorithms of the geolocation system.

Many approximation methods that are used to estimate the location of a signal from TDOA measurements are explored, but none fit the constraints of deploying the system onto a UAV or other smaller platform. A combination of these techniques is required to successfully locate low-power emitters, and will have to be developed to meet these strict limitations. Chapter III includes simulation methodology for a robust geolocation system that can locate a low-power emitter in the presence of multipath using multiple receivers, one of which only has a NLOS path to the emitter.

III. Simulation Methodology

3.1 Overview

This chapter provides a description of the simulations developed for this research to include the structure of the algorithms developed and used. An overall system description is presented followed by a detailed description of each system component and how each component is used in locating the emitter of interest.

3.2 Geolocation System

The proposed passive geolocation system is comprised of several parts: a) the sensors, b) the correlators, c) WLS position estimator, d) a NLOS sensor detection and identification algorithm and finally, e) a sensor redirection algorithm. The bulk of this research effort was spent on the identification and mitigation of NLOS sensors.

Figure 3.1 shows an example sensor and emitter configuration and the true and estimated positions of the emitter of interest. This is a simple case where each sensor has a LOS propagation path from the emitter and the sensors encompass the emitter.

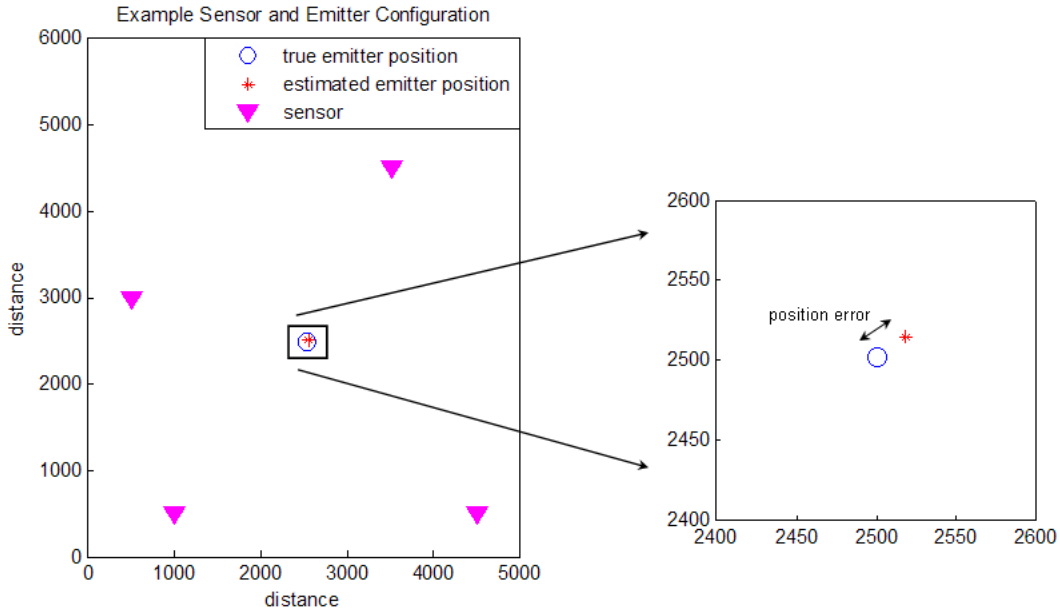


Figure 3.1: Example receiver configuration with estimated and true emitter positions. All sensors have a LOS signal path to the emitter.

In urban environments where buildings and towers are common, there is a need to identify when the LOS path from the emitter to the sensor is blocked. If a sensor's LOS path is blocked or reflected, the signal it receives is delayed, causing the time that the signal arrives at the sensor to be later than if the signal propagated directly to the sensor. This is the result of the signal reflecting off other buildings and obstacles before it reaches the receiving sensor. When this received signal is included in the geolocation solution, any TDOA calculation and resultant position estimate will include additional error. To estimate the position of the low-power emitter accurately, the source of this error needs to be determined and removed from the geolocation system. After the sensor is removed from the geolocation system, the emitter location can be re-estimated more accurately without the added NLOS error. Figure 3.2 shows a top-level view of the geolocation system.

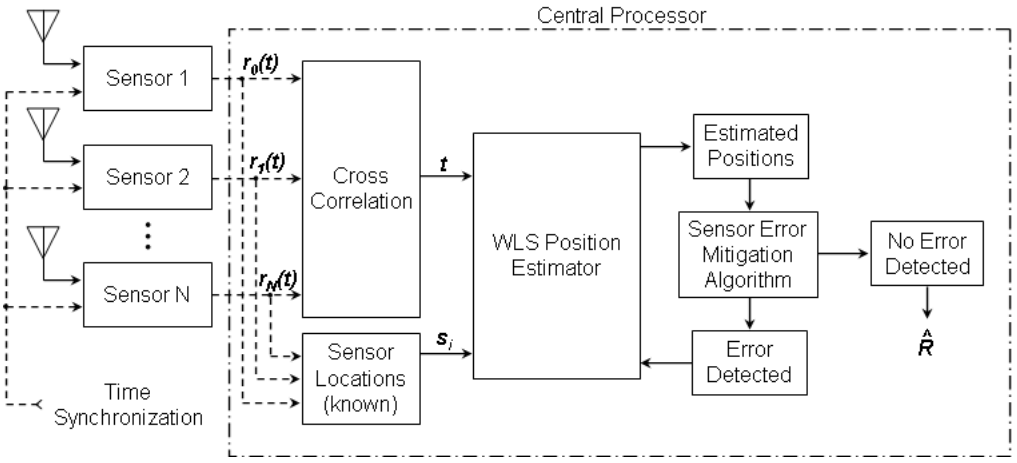


Figure 3.2: Passive geolocation system.

3.3 Sensor Parameters

The geolocation system includes N sensors that each receive the signal of interest. It is assumed that the sensors each have identical capabilities. Each sensor is omni-directional and can detect and receive the signal coming from the low-power emitter. All sensors have perfect timing and are synchronized.

The received signal and noise at each sensor is written as

$$r_i(t) = \alpha_i s(t - t_i) + n_i(t), \quad i = 1, 2, ..., N \tag{3.1}$$

where $i = 1$ refers to the reference sensor. The received signal, $\alpha_i s(t - t_i)$, is the original signal, delayed by $t_i = d_i/c$ seconds (the relative time it takes the signal to travel from the emitter to the i^{th} sensor), and attenuated by a factor α_i. $n_i(t)$ refers to the receiver noise in the i^{th} sensor. The receiver noise processes are assumed to be uncorrelated and independent, zero-mean, white Gaussian noises. It is also assumed that the noise power is consistent in all receivers (i.e., they have the same noise figure).

Each sensor down-converts, synchronously digitizes and time tags the received signal. Each sensor's location, its received signal, and timing information are all sent via wireless data link to a central processing location for near-real time processing. It is assumed that the central processor has unlimited access to each sensor and its data.

3.4 Cross-Correlation

Cross-correlation is completed by the central processor to determine the TDOAs between every pair-wise combination of received signals. Because the actual time the signal left the emitter is unknown, the time the signal arrived at each sensor is recorded and used as the time of arrival. This is also referred to as the relative TOA, annotated by t_i.

3.4.1 Correlation Technique. To find the TDOAs between the received signals at each receiver, Knapp's correlation technique described in Section 2.3.1 is used [15]. The signal data collected at each receiver is correlated with the signal data from every other receiver. Using Knapp's method, the generalized cross correlation function is used:

$$R_{xy}(\tau) = E[x(t)y^*(t - \tau)], \tag{3.2}$$

where $E[\,]$ is the expected value operator. This equation, assuming ergodicity, can be written as

$$\hat{R}_{xy}(\tau) = \frac{1}{t-\tau}\int_{\tau}^{T} x(t)y^{*}(t-\tau)dt, \tag{3.3}$$

where T is the observation interval. The value of τ that maximizes this cross correlation function is the TDOA estimate, $t_i - t_{i+1}$ [15].

The TDOAs between each pair-wise combination of receivers is estimated using the cross-correlation function in (3.3). Figure 3.3 displays the hyperbola set of solutions that can be drawn from calculated TDOAs between three simulated receivers located at "**x**." The estimated emitter location (*) lies at the intersection of the hyperbolas.

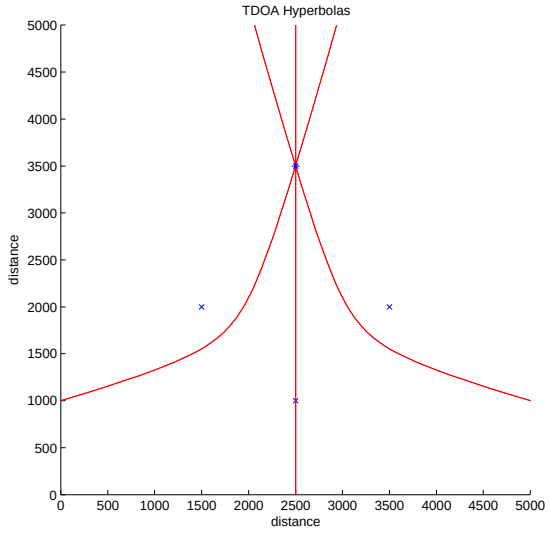


Figure 3.3: Estimated TDOA hyperbolas for three receivers located at "**x**."

3.4.2 Correlation Window. The correlation window size will affect the accuracy of the TDOA estimation. As Figure 3.4 shows, the peak of the autocorrelation of a modeled BPSK signal varies in width with respect to the correlation window. The BPSK signal parameters used in simulation are described in Section 3.7.1. Side lobes in the autocorrelation of a signal can indicate possible sources of error while

estimating the TDOA, especially in the presence of noise and interference. The more narrow the peak, the less room there is for error in determining the TDOA between two signals. But if the correlation window is too small, the correlation peak will be wider, and allow more error in any resulting estimate. The autocorrelation is also a function of the signal itself, so it also depends on the type of signal and its parameters.

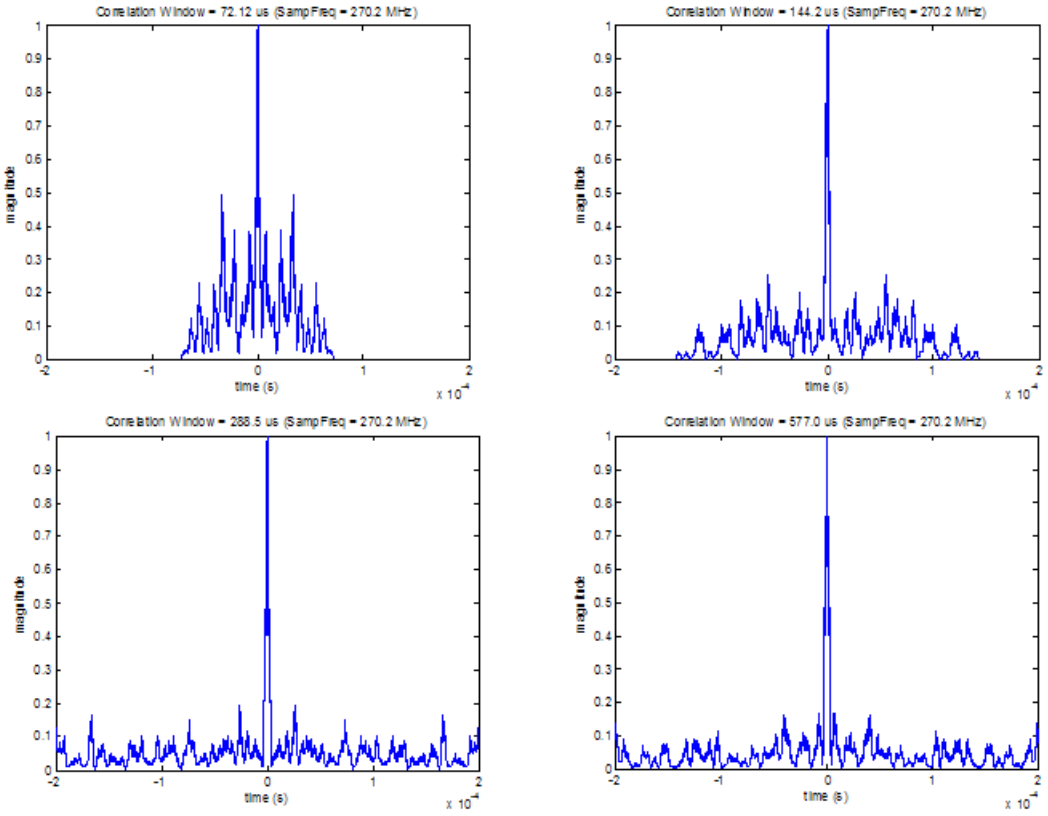


Figure 3.4: Autocorrelation of BPSK waveform signal applying different correlation window sizes (SNR=0 dB).

3.4.3 Sampling Frequency. A range of sampling rates are shown in Figure 3.5. As shown, when the sampling frequency is too low, the peak of the autocorrelation is relatively wide, leaving more room for error. The autocorrelation of the BPSK waveform signal varies little when sampled at 27.02 MHz or 270.2 MHz even though they differ by a factor of ten. In this case, the processor time needed to run ten times more data appears to yield little gain in resolution.

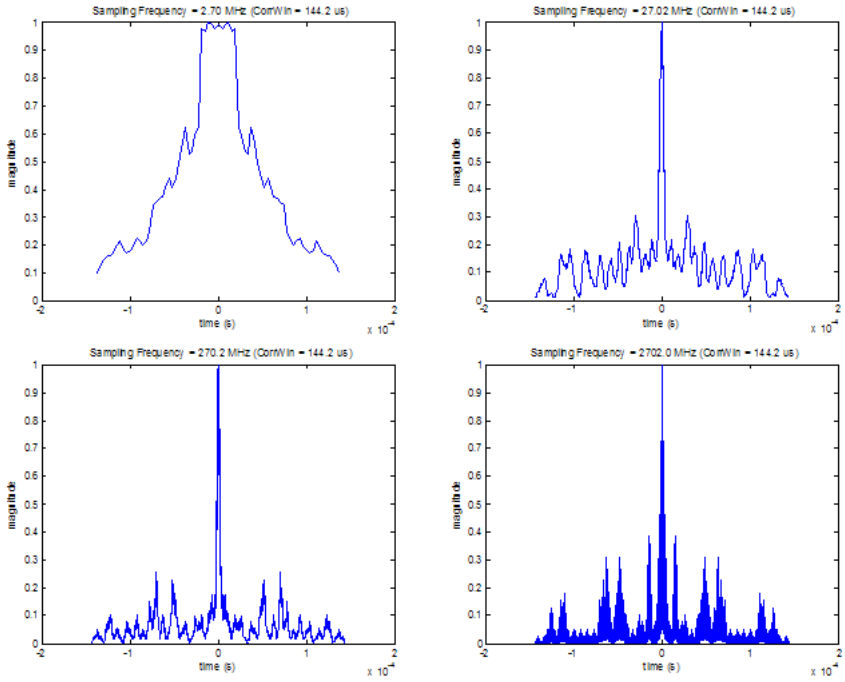


Figure 3.5: Autocorrelation of BPSK waveform signal after various sampling rates are used (SNR=0 dB).

3.4.4 Smoothing Function. Because there may be some noise in the signal, especially at lower power levels, the cross correlations are smoothed using a smoothing function. The smoothing function performs a moving average of a specified number of correlated points (N_s). An example of this can be seen in Figure 3.6 for $N_s = 8$. Eight was used as the moving average window because it produced results consistent with the desired ten meter estimated position solution resolution.

3.4.5 Resolution. As discussed in previous sections, precision of the estimated position is affected by the size of the correlation window and the rate of sampling. Increasing the correlation window size may increase the precision of the cross-correlation calculations, but a larger correlation window also adds more data points to process, adding to the overall processing time the system needs to estimate the position of the emitter. Increasing the sampling rate can also increase the resolution of the position estimate, but this also adds more processing time. Both methods

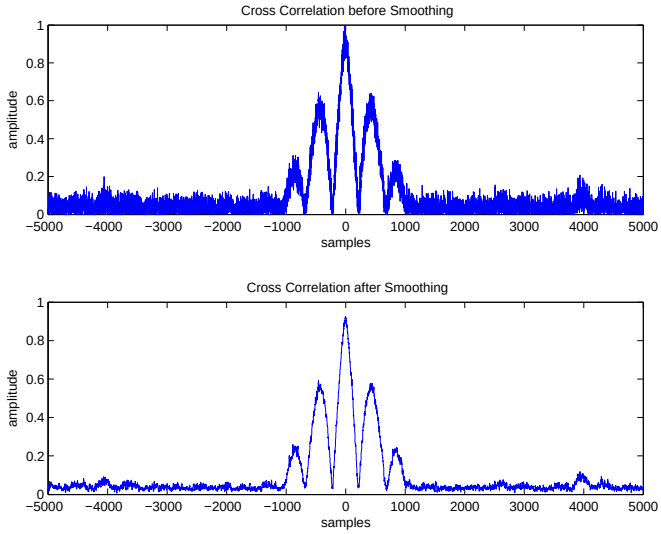


Figure 3.6: Smoothing function ($N_s = 8$): autocorrelation of BPSK signal before and after smoothing function is implemented.

of increasing the resolution of the estimate reach a point when adding more data points no longer adds to the precision of the estimate.

The correlation window size depends on the signal that is being correlated. For example, the GSM cell phone uses time division multiple access (TDMA) to optimize communication channels, where each time division is 577 μs long. To accommodate this, the GSM signal is sent in 577 μs long pulses every eight time divisions. For this case, using a correlation window longer than 577 μs would not be favorable because it would cause the geolocation system to spend unnecessary processing time correlating noise and interference while no signal is being sent. Section 3.7.2 provides more detail on the GSM cell phone signal characteristics.

For all results and analysis in this research, a correlation window of 144.2 μs is chosen to find each estimated TDOA. This is shorter than a GSM signal pulse but provides the desired resolution in the position estimate. Several different correlation windows and their effects on the estimated position variance and error of the emitter are explored in Chapter IV. The correlation window size and sampling rate will need

to be re-evaluated if the geolocation system is used to locate an emitter with a different signal waveform than is presented here.

The sampling rate can limit the resolution of the estimated position. For example, a sampling rate of 72.02 MHz results in 100 samples per 3.7 μs symbol, or 1 sample per 37 ns. If the signal travels at the speed of light, $c = 3.00 \times 10^8 m/s$, one sample will be taken every 11.1 meters. This corresponds to the geolocation system having a best possible resolution of 11.1 meters at that sampling rate. Table 3.1 includes possible sampling rates and their impact on the estimated position resolution.

Table 3.1: Signal sampling parameters.

Samples/ Symbol	Sampling Interval (nsec)	Sampling Frequency MHz	Possible Resolution (m)
10	369	2.70	111
20	185	5.41	55.6
40	92.3	10.8	27.7
100	36.9	27.1	11.1
200	18.5	54.0	5.54

The geolocation system needs to have the ability to locate an emitter in near-real time while keeping the estimate as accurate as possible. Therefore a balance between the resolution of the position estimate and the time required by the processor to gain that resolution needs to be met.

3.5 Position Estimation Using WLS Estimator

A method of estimating the location of a signal source from TDOA measurements is presented that takes advantage of extra receiver TDOA measurements. This method detects and identifies sensors that provide poor relative TOA measurements due to NLOS error or low signal power. This is done by using a combination of several proven location estimation schemes including Residual Ranking and a Weighted Least Squares Estimation method that were detailed in Chapter II [6, 7].

The TDOA measurements found previously through cross correlation are used to estimate a location. As Figure 3.2 showed, the TDOA measurements between the sensors along with an initial guess of the emitter position are inputted into Torrieri's WLS location estimation algorithm [28]. Figure 3.7 displays the geometrical configuration referred to by the WLS position estimator.

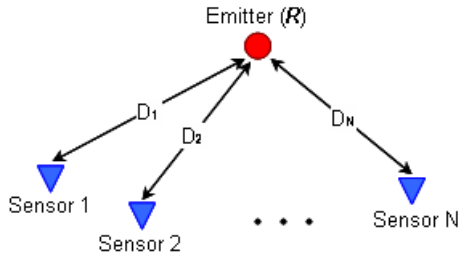


Figure 3.7: Geometry of emitter and N sensors [28].

This algorithm, like most other estimation algorithms, needs an initial position guess to estimate the position from TDOA measurements. Because the proposed geolocation system is passive, it is not known initially what area the emitter may be in, so the centroid, or arithmetic mean between the receivers, $\boldsymbol{R}_0 = [\bar{x},\ \bar{y},\ \bar{z}]$, is used as the initial position guess.

The distance from sensor i, $\boldsymbol{s}_i$, to the reference location, $\boldsymbol{R}_0$, is defined as the scalor

$$D_{0i} = \|\boldsymbol{R}_0 - \boldsymbol{s}_i\|, \tag{3.4}$$

where $\boldsymbol{s}_i$ is the matrix containing the known locations of each sensor. The matrix of derivatives evaluated at the coordinates of the reference location, $\boldsymbol{F}$, is calculated from (3.4) as

$$\boldsymbol{F} = \begin{bmatrix} (\boldsymbol{R}_0 - \boldsymbol{s}_1)^T / D_{01} \\ (\boldsymbol{R}_0 - \boldsymbol{s}_2)^T / D_{02} \\ \vdots \\ (\boldsymbol{R}_0 - \boldsymbol{s}_N)^T / D_{0N} \end{bmatrix} \tag{3.5}$$

By estimating the TDOAs between each sensor, the initial time the signal left the emitter, t_0, is not needed. The TDOA measurements are written as

$$t_i - t_{i+1} = (D_i - D_{i+1})/c + n_i, \quad i = 1, 2, ..., N, \tag{3.6}$$

where n_i is the measurement error and $t_i - t_{i+1}$ is the difference in time the signal arrived at each i^{th} sensor. To simplify calculations in Matlab®, (3.6) is written in matrix form

$$\boldsymbol{Ht} = \boldsymbol{HD}/c + \boldsymbol{H}\epsilon, \tag{3.7}$$

where $\boldsymbol{t}$, $\boldsymbol{D}$, and ϵ are N-dimensional column vectors with components t_i, D_i and ϵ_i. ϵ refers to the arrival time measurement error, which accounts for propagation anomalies, receiver noise, and errors in the sensor positions. The $\boldsymbol{H}$ matrix, $(N-1) \times N$ is given as [28],

$$\boldsymbol{H} = \begin{bmatrix} 1 & -1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -1 \end{bmatrix}.$$

The final equation for the weighted least squares estimator then becomes [28]

$$\hat{\boldsymbol{R}} = \boldsymbol{R}_0 + c(\boldsymbol{F}^T\boldsymbol{H}^T\boldsymbol{N}^{-1}\boldsymbol{HF})^{-1}\boldsymbol{F}^T\boldsymbol{H}^T\boldsymbol{N}^{-1}(\boldsymbol{Ht} - \boldsymbol{HD}_0/c), \tag{3.8}$$

where $\boldsymbol{N}$ denotes the covariance matrix of the arrival-time errors. It is defined as

$$\boldsymbol{N} = \boldsymbol{HN}_\epsilon\boldsymbol{H}^T. \tag{3.9}$$

where $\boldsymbol{N}_\epsilon$ represents the estimated noise in the receivers,

$$\boldsymbol{N}_\epsilon = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \sigma_N^2 \end{bmatrix}, \tag{3.10}$$

and σ_i^2 is the arrival-time variance.

During each iteration of the LS algorithm, the position estimate and TDOA estimates are used to generate a new position estimate for the emitter. This position estimation method continues until the position estimate converges to a location or the algorithm loop is stopped. To determine if the position estimate has converged to a solution, the standard deviation, s, is used

$$s = \sqrt{\frac{1}{n-1}\sum_{j=1}^{n}(\hat{\boldsymbol{R}}_j - \bar{\boldsymbol{R}})^2}, \quad j = 1, 2, ..., n \tag{3.11}$$

where $\hat{\boldsymbol{R}}_j$ refers to the j^{th} iteration of the position estimate, and $\bar{\boldsymbol{R}}$ refers to the mean of the estimated locations.

Initial runs of the WLS estimator shows that when the estimate converges to a solution, it does so over just a few iterations of the estimator. By comparing the last three estimates the WLS algorithm produces, it can be determined if the algorithm has converged to a solution, or if more iterations of the WLS estimator are needed. This saves precious processor time by eliminating redundant position estimates. Comparing more than the last three estimates yields a position estimate with no less error than comparing just the last three, and takes more time to run since more calculations are needed. Standard deviation is used to compare the last three position estimates. If it is found that the position estimates are within ten meters of each other, it is assumed that the solution has converged, the WLS estimator is stopped, and the last position estimate is used as the final estimate out of the WLS

estimator. Ten meter resolution is consistent with the initial standards set for this research in Chapter I. Depending on where the emitter is in relation to the sensors, it may take several iterations of the WLS estimator to converge to a solution. An example of an estimated location using this method is shown in Figure 3.8. In this case, it took five iterations of the WLS location estimator to converge to a solution close to the true location of the emitter.

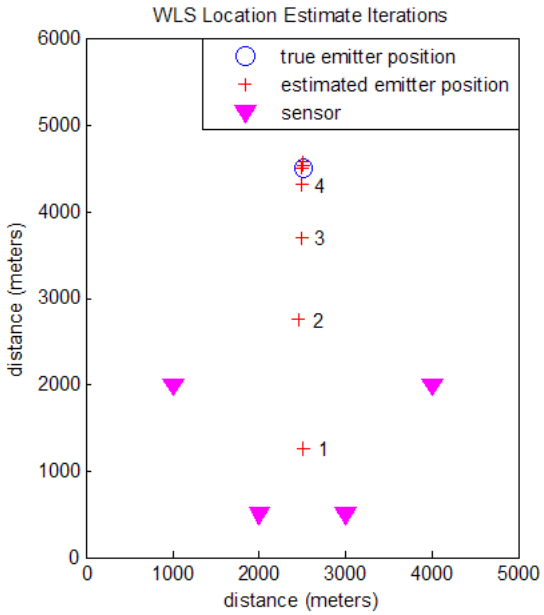


Figure 3.8: Estimated position using Torrieri's method [28]. The numbered estimated positions correspond to the order of iterative emitter location estimates calculated, eventually converging close to the true emitter location (SNR = 0).

3.6 Detection and Identification of NLOS Receiver(s)

As mentioned throughout this research effort, there are many challenges in locating a emitter like a cell phone or walkie-talkie in an urban environment. Besides the low-power signal levels involved, there are buildings, highways, and towers that could be in the way of a sensor that is trying to detect and locate the emitter. Since it is not known exactly where the emitter of interest is, the geolocation system must be

able to determine if one of its sensors does not have a LOS propagation path from the emitter. Therefore, a proposed method of identifying a NLOS receiver is presented.

3.6.1 Approach. Figure 3.9 displays the flow chart for this segment of the geolocation system.

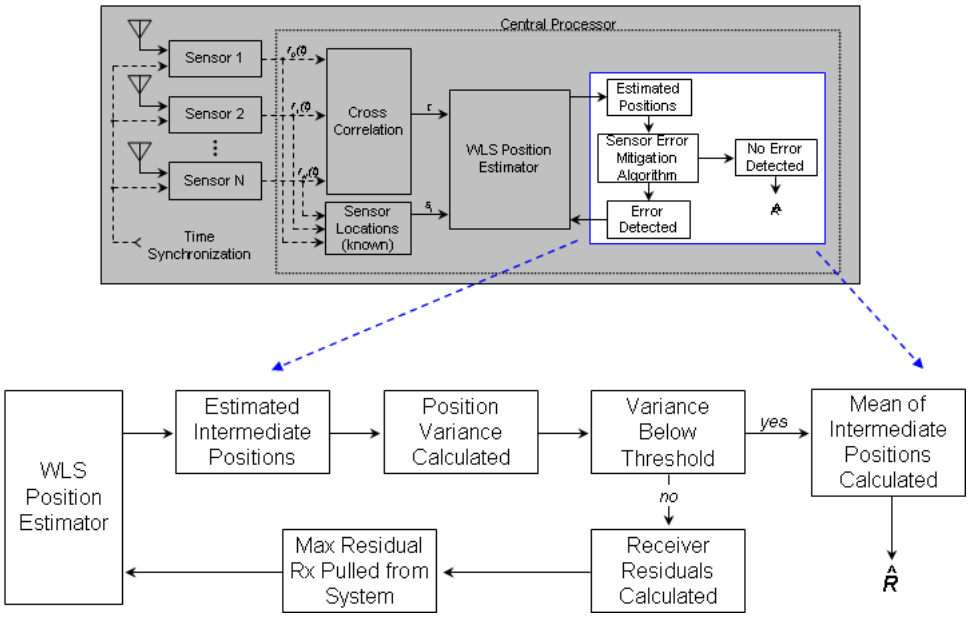


Figure 3.9: Bad sensor detection and identification process.

The equation

$$N - K \geq L, \tag{3.12}$$

is used to describe how many LOS and NLOS sensors there are, where N is the total number of sensors, K is the number of NLOS sensors, and L is the minimum number of LOS sensors required to estimate a position. In 2D position estimation, $L = 3$ LOS sensors are needed to estimate a position.

Assuming there are sufficient sensors that have a LOS propagation path to the emitter, just one emitter position estimate calculated with the WLS estimator is needed for the Sensor Error Mitigation Algorithm to determine if a sensor adds too much variation to the estimated emitter position. The sensor may only have a NLOS path from the emitter and needs to be moved to another location with a LOS path

to the emitter to further optimize the position estimate. After the K NLOS sensors are removed from the computations, an improved emitter location estimate can be calculated using only the $N - K$ LOS sensors, assuming there still remains at least L sensors remaining.

3.6.2 Sensor Error Mitigation. Adding more receivers to the geolocation system does not always increase accuracy in the emitter location estimate. If a receiver is added to the system, but has only a NLOS signal propagation path to the emitter, more processor time is needed to identify this bad receiver and to omit it from the geolocation system. The effects of propagation also play a role in increasing the amount of error due to the added noise and decrease in signal strength.

It is assumed for the scope of this research that the number of sensors with a LOS signal path to the emitter is equal to or greater than the minimum required to estimate a location, L. As mentioned earlier, to determine a two dimensional emitter location, the minimum number of sensors needed is $L = 3$. If there are more sensors than are minimally required to estimate a location ($N > L$), there is more flexibility in how the TDOA calculations are grouped and used in the WLS location estimator. For example, if there are $N = 5$ sensors, there are 16 different combinations that can be used to estimate the emitter location.

1. Select 5 out of 5: $\binom{5}{5} = 1$ possible combination
2. Select 4 out of 5: $\binom{5}{4} = 5$ possible combinations
3. Select 3 out of 5: $\binom{5}{3} = 10$ possible combinations

Applying the WLS estimator to these combinations results in 16 different emitter location estimates. These estimates are referred to as intermediate location estimates.

Because these intermediate location estimates result from different combinations of sensor TDOA calculations, each estimate will have varied weights of each sensor's relative TOA information contributing to it. If there is a NLOS sensor, some of the position estimates will contain NLOS errors, and others may have fewer or no NLOS

errors. These errors depend on which sensor has the NLOS signal path from the emitter and can result in measurable variance in the estimated intermediate positions.

3.6.3 Emitter Position Variance. If the number of sensors in the geolocation system is small, a NLOS sensor may cause more error and variance in the estimated position of the emitter than if the number of sensors is large. For example, if there are $N = 5$ sensors, and 1 sensor has a NLOS propagation path to the emitter, the NLOS sensor will cause there to be some variation and error in the position estimate. But if there are only $N = 3$ sensors, and 1 is a NLOS sensor, the variance of the resulting position estimate is expected to be larger than in the $N = 5$ case described.

3.6.3.1 Variance Calculation. A plot of the intermediate estimated positions of a geolocation system that has a NLOS sensor is shown in Figure 3.10. Each intermediate position plotted represents the result of the WLS estimator after three sensors and their respective TDOA calculations are used. The variance of these intermediate location estimates is calculated and plotted for different cases to determine what the variance is when there is a NLOS sensor in the system.

By using only L sensors' relative TOA measurements in the WLS algorithm, the time needed to calculate the position estimate is much less because of the decrease in matrix calculations needed. If there are $N = 5$ sensors, as mentioned in Section 3.6.2, there is just one possible way to select all five sensors at once, and ten possible ways to select three sensors. Using the WLS method described earlier to estimate the emitter's position, one position can be estimated using all five sensor locations and their respective TDOA information, while ten positions can be estimated by using each combination of three sensors, $\binom{5}{3}$, and their respective TDOA information into the WLS estimator.

Although it may seem to be more efficient to only run the estimator one time, inputting 5 arguments 1 time takes approximately the same amount of time to estimate a position as inputting 3 arguments 10 times. The ability to run more WLS estimates

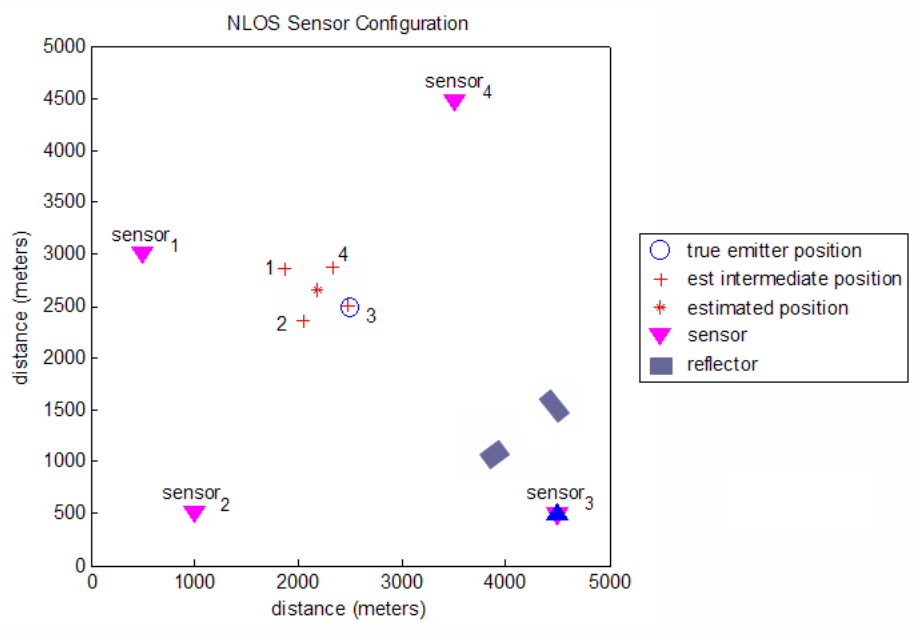


Figure 3.10: Sensor configuration with 3 LOS receivers and 1 NLOS receiver. Numbered intermediate position estimates correspond to the estimate after that respective sensor is removed from geolocation system. The final estimated position is a result of the mean of all intermediate position estimates.

with fewer input arguments allows the NLOS sensors to have a greater impact on the position estimates, allowing them to be easier to detect and identify.

The variance of the intermediate position estimates is calculated using the equation

$$s_{n-1}^2 = \frac{1}{n-1}\sum_{j=1}^{n}(r_j - \bar{r})^2, \quad j = 1, 2, ..., n, \tag{3.13}$$

where n refers to the number of position estimates. The actual emitter location is unknown, so the position mean, referred to in (3.13) as $\bar{r}$, is used. The mean square error (MSE) is calculated using the following equation

$$MSE = \frac{1}{n}\sum_{j=1}^{n}|e_j|^2, \quad j = 1, 2, ..., n, \tag{3.14}$$

where $e_j = R - \hat{R}_j$, and j corresponds to the j^{th} position estimate.

It is assumed when a NLOS sensor is included in the estimation of the emitter position, the variance and error is much greater than if all sensors have a LOS signal propagation path to the emitter. The variance of the intermediate position estimates is compared to the variance of the estimated location when there is no NLOS sensor. If the variance of the estimated location is higher than the variance of the estimated position with no NLOS sensor, then it is assumed that there is at least one receiver that has a NLOS path to the emitter, adding error to the location estimate.

3.6.3.2 Sensor Residual. If the variance of the final position estimate is above a desired threshold set, the variance of the position estimates with respect to each sensor is calculated using (3.13). In the case of $N = 5$ sensors, each individual sensor has $\binom{5-1}{3-1} = 6$ possible combinations of $L = 3$ sensors that include itself, that the central processor can use to estimate a position. These six position estimates are separated from the geolocation system, and the variance is determined. This variance, now termed sensor residual, is compared with the residual of all the other respective sensor position estimates. The steps taken by the geolocation system to find the residual for a sensor's position estimate are detailed below:

1. Use all pairwise-combinations $\binom{N-1}{L-1}$ of TDOA measurements with respect to each sensor i to estimate the position. Estimates using the same combination of sensors have already been calculated by the central processor.
2. Calculate the variance of these estimated positions. This is referred to as sensor i's residual.
3. Complete step 1 and step 2 for every i^{th} sensor.
4. Rank the sensor residuals from 2.

An example of $N = 5$ sensors with 1 NLOS sensor is shown in Figure 3.11. Each plot shows the intermediate estimated emitter positions with respect to each of the reference sensors highlighted. The variance of the estimated positions in each plot correspond to the residual variance for that respective sensor. Figure 3.11c shows

the most variance in the estimated positions because each position estimate has the NLOS sensor error contributing to it. The other plots show less variance in the position estimates because not all the intermediate position estimates have the NLOS sensor error.

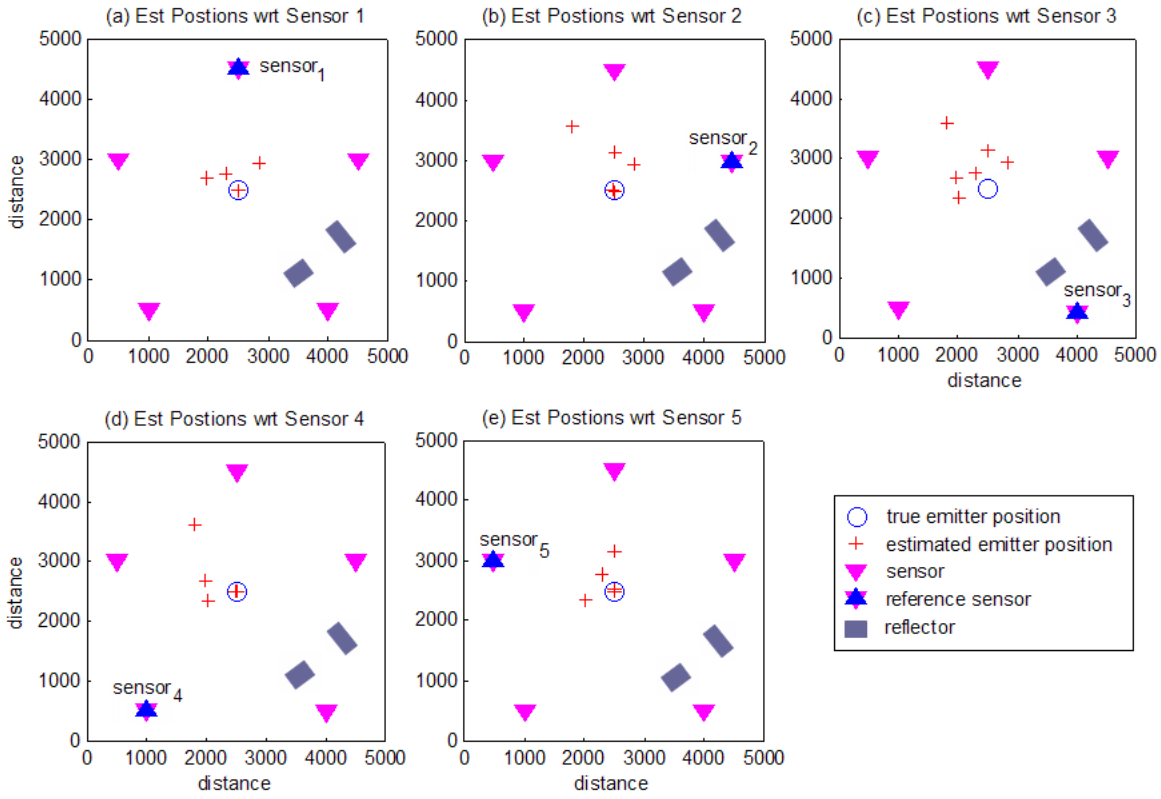


Figure 3.11: Plots of emitter position estimates with respect to each reference sensor. Sensor 3 has a NLOS path to the emitter.

The residual refers to the variance that each sensor's TDOA measurements bring to the overall position estimate. The sensor with the largest residual is removed from the location estimation process, since this sensor's probability of being the NLOS sensor is the highest. The emitter's location is re-estimated with the remaining sensors. If the variance of the location estimates falls below the desired threshold after removing the receiver from the system, it is determined that sensor has only a NLOS path from the emitter, and created the additional error. If the variance of the location estimate remains above the desired threshold, the sensor previously removed from the

system is added back in, and the sensor with the next largest residual is removed from the system. This process of isolating each sensor one at a time is continued until one is identified as the bad sensor, or all sensors have each been pulled individually from the system.

If the variance remains above the desired threshold even after each sensor has been removed from the algorithm, then each pair-wise combination of receivers is taken out. This continues until the variance of the location estimate falls below the threshold level set, or there are just L receivers left (the minimum number of receivers needed to calculate the location of an emitter using TDOA). If there is no sensor determined to be the main error contributor at this point, it is determined the geolocation system cannot accurately estimate the emitter's position and produces no result.

The reflectors modelled during this simulation are stationary, delaying the relative TOA of the signal that it comes in contact with the same during each trial. Because TDOA is the geolocation method used, any phase shift associated with the addition of reflectors is ignored. It is expected that the variance will be consistent and predictable for each of the sensor configurations being tested. Thresholds will be set so that future position estimate variances can determine if there is a NLOS sensor.

3.6.4 Recommended Sensor Locations. After the emitter location is estimated by the methods described previously, the sensors identified as NLOS can be moved to a better location. This may be especially useful if the geolocation system is deployed onto UAVs or UGVs that are remotely controlled, and can be moved to a different location on demand. Chapter II discussed Drake's research into control of multiple UAVs for passively locating radars [8]. For the 2D case, we can write the sensor new location as

$$x_{i.new} = x_i + \triangle x_{adj}, \quad i = 1, 2, ..., N \tag{3.15}$$

$$y_{i.new} = y_i + \triangle y_{adj}, \quad i = 1, 2, ..., N \tag{3.16}$$

where $(x_i,\ y_i)$ is the original location of the i^{th} sensor, $(\triangle x_{adj},\ \triangle y_{adj})$ is the difference between the old and new location of the i^{th} sensor, $(x_{i.new},\ y_{i.new})$ is the new sensor location, and $\triangle$ can be either positive or negative.

If a sensor is determined to have a NLOS propagation path to the emitter, an adjustment to the location of the sensor is made. Because the characteristics of the obstacle blocking the LOS path are not known, the NLOS sensor is directed to move in a direction perpendicular to the line between the emitter and sensor while maintaining the same distance to the emitter. If it still has a NLOS path to the emitter, the sensor continues moving the same direction until the blockage no longer has an effect on the emitter's signal.

3.7 Emitter Signal Generation

Two types of signals are used during simulation and analysis of the geolocation system. The first signal is a common, relatively simple binary phase shift keyed (BPSK) signal. This signal has known characteristics including autocorrelation and power-spectral density, providing a good baseline to test the geolocation system prior to simulating the more complex gaussian minimum shift keyed (GMSK) waveform signal. This section provides more detail on how these two modelled signals are generated.

3.7.1 Baseline Signal: BPSK Waveform Signal. Parameters similar to those in a GSM cell phone signal are applied for consistency and easier analysis and comparison. The bits used are independent and randomly generated using the "rand" command in Matlab®. These are (BPSK) modulated using 1 bit/symbol. The bit rate applied is approximately 270 kHz which is a result of a symbol period of 3.7 μs. The signal is set in the base band to reduce complexity of the signal. The initial phase is set to zero.

3.7.2 Test Signal: GMSK Waveform Signal. As Chapter II mentions, the GSM cell phone can be an easy tool for potential terrorists to use because of its popularity and versatility. The GSM signal is used as the representative waveform during this research, but the simulation can be expanded to include other emitters of interest. While the need still exists to locate an emitter while in the presence of other interfering signals, for the purposes of this research effort, it is assumed that only one stationary emitter exists in the area of interest.

A gaussian minimum shift keyed (GMSK) waveform signal, modeled after a common global system for mobile communications (GSM) cell phone signal, is simulated and used as the test signal. The Matlab® code for the GMSK waveform signal generation is adopted from Sikes' research [23]. Taking characteristics from an actual recorded GSM signal from Marino's work [18], the Matlab® code is developed to simulate a GSM signal that is emitted from an active GSM cell phone. This allows the levels of noise and attenuation to be modelled for the simulation.

It is observed from Marino's work that the emitted GSM signal is made up of series of four pulse trains, each 18.5 ms in length. This corresponds to four TDMA frames being utilized in a row. The structure of a GSM frame is shown in Figure 3.12. Each individual pulse is made up of 148 randomly generated bits that are GMSK modulated. A pulse is 577 μs long in time and takes up one TDMA slot of time. A series of the short burst pulses are shown in Figure 3.13. It is assumed that each sensor can detect when a pulse begins, independent of any noise that may be present in the receiver or data channel. After it is determined that every sensor in the system can detect the signal coming from the low-power emitter, all sensors are instructed by the central processor to record and send the next pulse for analysis. At this time, each i^{th} sensor waits until it detects the next pulse in the signal. It then time stamps the signal and records 577 μs of the GSM signal to send to the central processor.

This time period is equal to the maximum correlation window length that is used by the central processor to correlate and determine the TDOAs between the

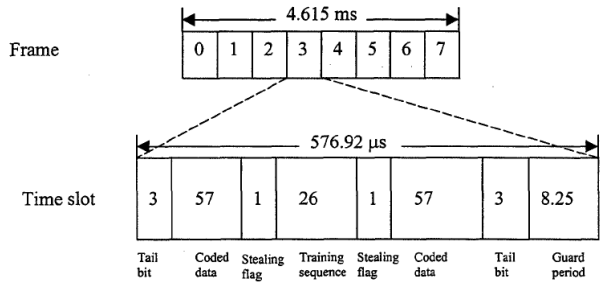


Figure 3.12: GSM frame details [21].

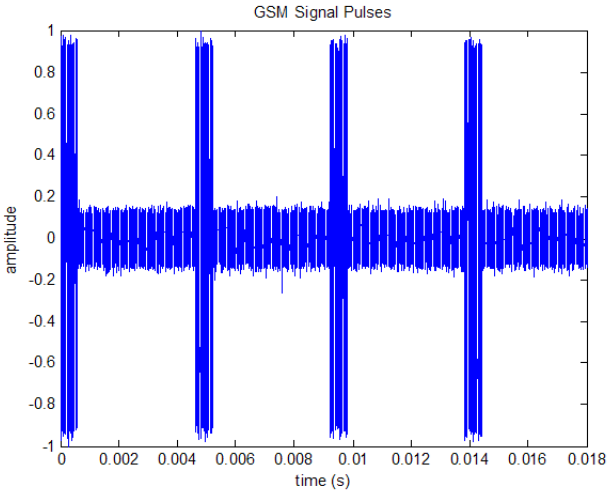


Figure 3.13: Series of simulated GSM signal pulses. Each pulse corresponds to an assigned time slot. Pulses are later referenced as pulse 1 through pulse 4.

received signals. By only sending 577 μs of data to the central processor, the amount of data, and therefore bandwidth needed, is minimized. In this method, no extra data points are sent to the central processor, saving valuable processor time and bandwidth. Because each signal pulse is independent, if a sensor accidentally records and sends the wrong pulse to the central processor, the central processor will easily detect the error after that pulse is cross-correlated with another pulse received at a different sensor. This is shown in Figure 3.14. The remaining parameters are set the same as in the BPSK signal model described previously.

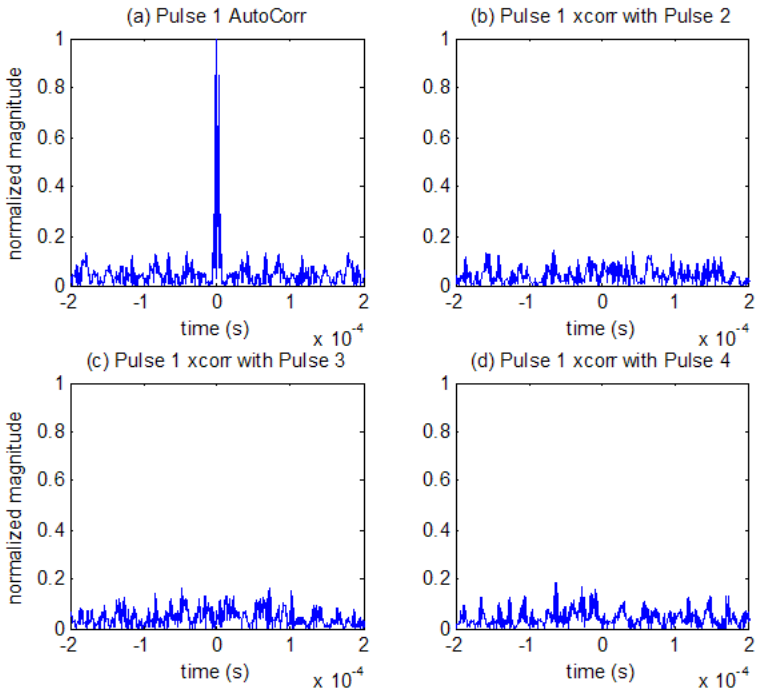


Figure 3.14: Cross-correlation of GSM signal pulses. The autocorrelation of pulse 1 is compared with the cross-correlation of pulse 1 with pulse 2, pulse 3, and pulse 4. All plots are normalized to the maximum value of the autocorrelation in (a).

3.7.3 Noise Generation. The noise added to the modeled signals is assumed to be additive white Gaussian noise (AWGN) and uncorrelated. Even though the receivers are assumed to be identical, they are all physically different systems, with different noise variations. The noise amplitude is the same for each receiver. This amplitude is calculated from the signal-to-noise ratio (SNR) value that is specified. For this research, SNR is calculated as

$$SNR = \frac{Average\ Received\ Signal\ Power}{Average\ Received\ Noise\ Power} \tag{3.17}$$

The actual noise and signal amplitudes used are calculated from the SNR values simulated.

3.8 Urban Environment Generation

To simulate the effects of the urban environment in Matlab®, reflectors, like buildings and towers, are added. When the distance from a sensor to the true location of the emitter is calculated, the distance from the sensor to the reflector is added to the distance from the reflector to the emitter. This simulates the added distance and time the signal will travel on its way to the sensor. As illustrated in Figure 3.15, the distance between the emitter and Sensor 3 would be equal to $D_{3a} + D_{3b}$ and not equal to the direct difference in distance between the emitter and sensor.

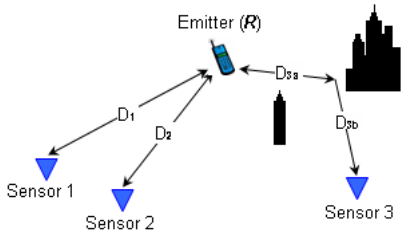


Figure 3.15: Simulated signal propagation path block, reflector and its effects on the distance the signal travels from the emitter to each sensor.

3.9 Summary

This chapter includes an overview of the Passive Geolocation System. This system provides a robust way to passively geolocate a low-power emitter while in the presence of multipath and noise. The possible outcomes include (1) correct location of the emitter, (2) incorrect location of the emitter, and (3) unknown no location of the emitter.

Chapter IV includes results and analysis of the geolocation system with the test signals developed in Chapter III. Predictable position variance and error for different sensor configurations, from simulating the proposed geolocation system using Matlab® are presented.

IV. Geolocation Results and Analysis

4.1 Overview

This chapter presents the results and analysis of the modeled geolocation system performance as simulated using Matlab®. Position estimate analysis and the effects of various sensor configurations are provided. Baseline results are presented from running the geolocation system to estimate the location of a BPSK emitter with several different sensor configurations. Analysis of the overall problem, locating an emitter in the urban environment, is also provided.

Although this method can be applied to 3-dimensional space, all simulations are based on 2-dimensional configurations to simplify the simulations and analysis. To obtain a position estimate in two dimensions, three sensors are needed. Each simulation contains 500 Monte Carlo trials. For a given simulation, each trial maintains the same configuration of sensors, emitter, and reflectors, but differs in the noise realizations and the randomly generated binary data that is modulated to produce each BPSK signal. The sensor, emitter, and reflector locations are based on a local coordinate system that is measured in meters. Specific simulated sensor configurations are shown followed by their corresponding results.

4.2 System Results and Analysis: Baseline BPSK Signal

As Section 3.7.1 mentions, the BPSK waveform is a commonly used signal, with known characteristics that make it a good signal to baseline the proposed geolocation system with and verify the system can successfully locate a low-power emitter in a simulated urban environment. The BPSK signal, with applied parameters similar to those of a GSM signal, is used to provide baseline results for this geolocation system.

4.2.1 BPSK Signal Precision Characteristics. Although the geolocation system can be used to estimate the location of many different types of emitters, the error in the resultant position estimate is dependent on the width of the signal's autocorrelation function. The autocorrelation peak width has a large impact on the

variance and error of the estimated position of the low-power emitter produced by the geolocation system.

The BPSK signal and its respective autocorrelation are shown in Figure 4.1. The BPSK signal has a relatively narrow autocorrelation, which corresponds to less error when the TDOA is found via cross-correlation. If the autocorrelation were wider there would be increased error, especially when receiver and channel noise are added to the signal.

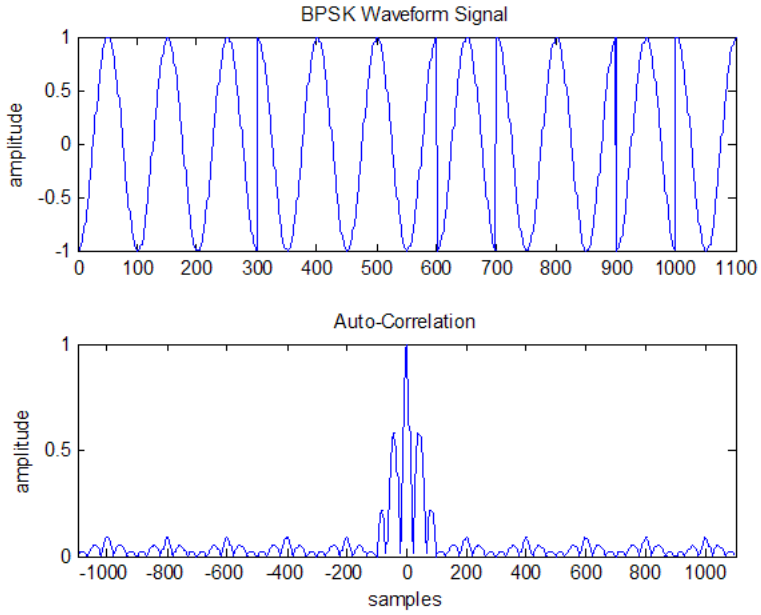


Figure 4.1: Simulated BPSK signal. Top figure is a BPSK waveform signal modulated from the 11-bit Barker sequence, bottom figure shows the absolute value of its autocorrelation. Number of samples per symbol is 100.

4.2.2 Scenario A: Effects of Data Rate on Position Estimate. For this research, a favorable sensor alignment is defined as the case when all sensors are located at equal distances from the emitter and distributed evenly in azimuth around the emitter. This alignment with five LOS sensors is used to show how different sampling rates can effect the error and variance of the position estimate even in a favorable sensor configuration.

In this scenario, the sensor and true emitter locations are as shown in Figure 4.2. The average over 200 trials of the estimated positions (denoted with a "*") resulting from each sample rate fell very close to the true position of the emitter. While the sensors encompass the emitter, it was found that the error in the estimated position caused the average position over those trials to converge to the true emitter location. The variance in position estimates and average error versus SNR are shown in Figure 4.3 for varying data rates.

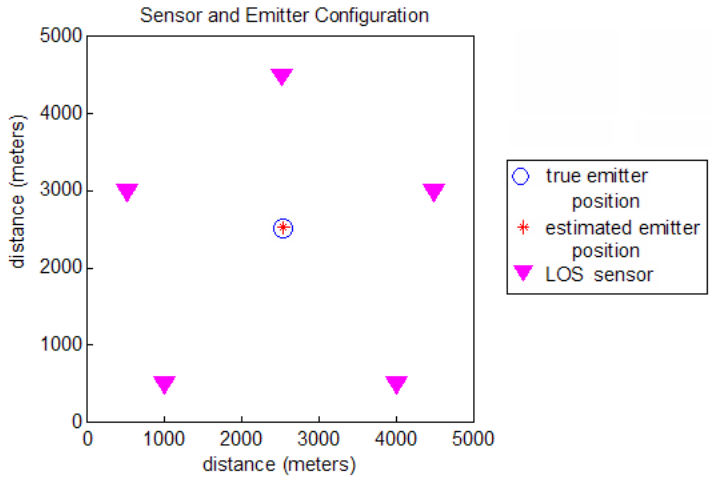


Figure 4.2: Favorable LOS configuration. The position estimate has very little variation and the error is minimized. Distance is calculated in meters ($SNR = 0\ dB$).

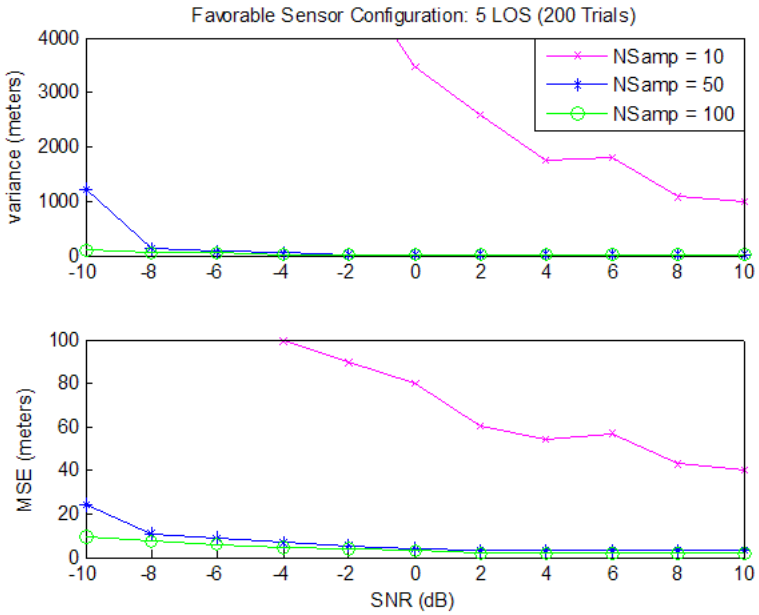


Figure 4.3: Data rate comparison results. The variance and error in the position estimate visibly improve with higher data rates.

4.2.3 Scenario B: Effects of Correlation Window on Position Estimate. A favorable sensor alignment is again used to show how different length correlation windows can effect the error and variance of the position estimate even in a favorable sensor configuration.

In this scenario, the sensors and true emitter location are positioned as shown in Figure 4.2. The estimated positions resulting from each correlation window size used is also shown. 200 trials of each data rate were run. The variance in the position estimate and error are shown in Figure 4.4.

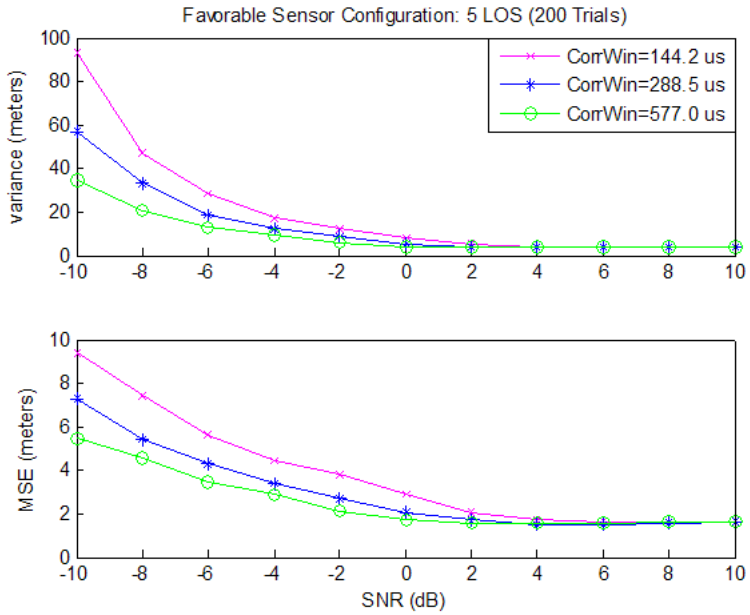


Figure 4.4: Correlation window comparison results. The variance and error in the position estimate improve with longer correlation window lengths.

4.2.4 Scenario C: Favorable Sensor Configuration. For this research, a favorable sensor alignment is defined as the case when all sensors are located at equal distances from the emitter and distributed evenly in azimuth around the emitter. In this scenario, the sensors, emitter, and reflectors are again positioned as shown in Figure 4.2. The estimated position resulting from this configuration is also shown and results from 500 trials of each sensor configuration. The variance in the position estimate and error in Figure 4.5 are used as a reference for comparison later to detect when there is a NLOS sensor in the geolocation system. A correlation window of 144.2 *us* and a sampling rate of 100 samples per waveform were chosen. These provided the position resolution needed while not burdening the central processor with too many computations.

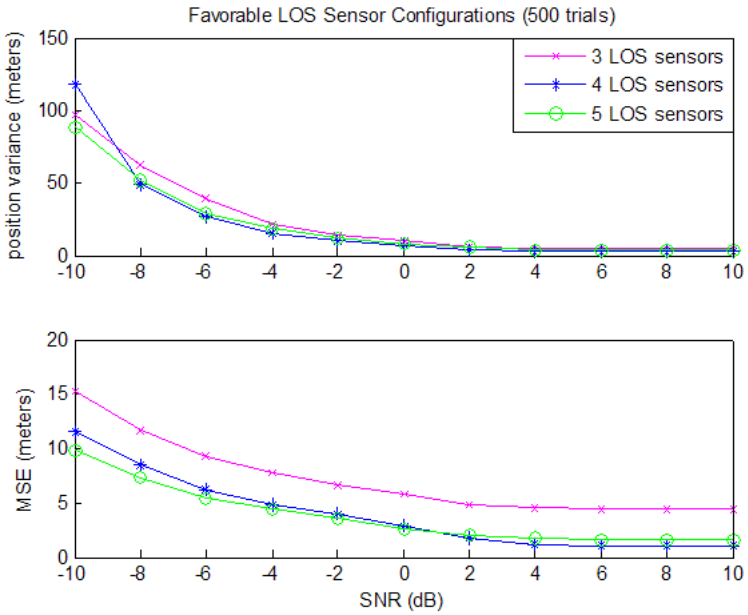


Figure 4.5: Favorable LOS configuration data. The variance and error is minimal while in this configuration.

4.2.5 Scenario D: NLOS Sensor Configuration. In this scenario, reflectors are added to the favorable sensor configuration of Scenario C to simulate buildings and towers that may block the emitter LOS path to a sensor. This configuration is shown in Figure 4.6. The resulting position errors and variance in this configuration at different SNR values are shown in Figure 4.7.

4.2.5.1 Detection and Mitigation of NLOS Sensor. The results in Figure 4.7 show that when a NLOS sensor is included in the geolocation system, the variance and error in the estimated position are much higher than when all sensors have a LOS to the emitter. This result shows that position estimate variance can provide an indication if there is a NLOS sensor present in the system. The geolocation system uses this higher variance to detect when there is a NLOS sensor.

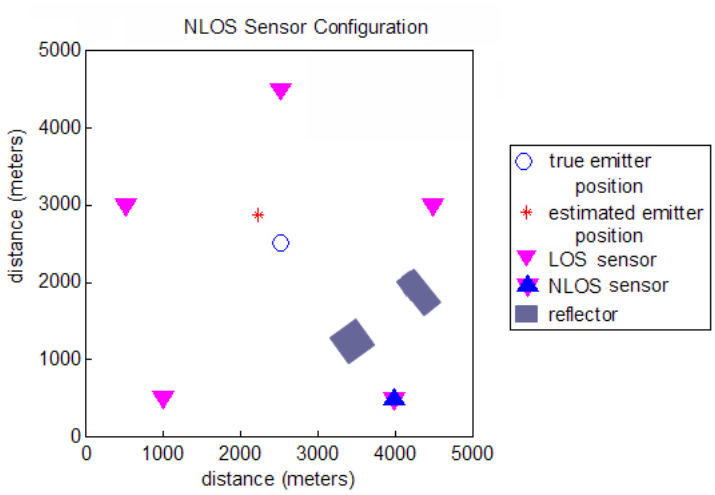


Figure 4.6: NLOS sensor configurations used in simulation. There are 4 LOS and 1 NLOS sensors. The estimated emitter position's error is a result of all sensors TDOA information (including the NLOS sensor) included and used by the central processor to estimate the position of the low-power emitter ($SNR = 0\ dB$).

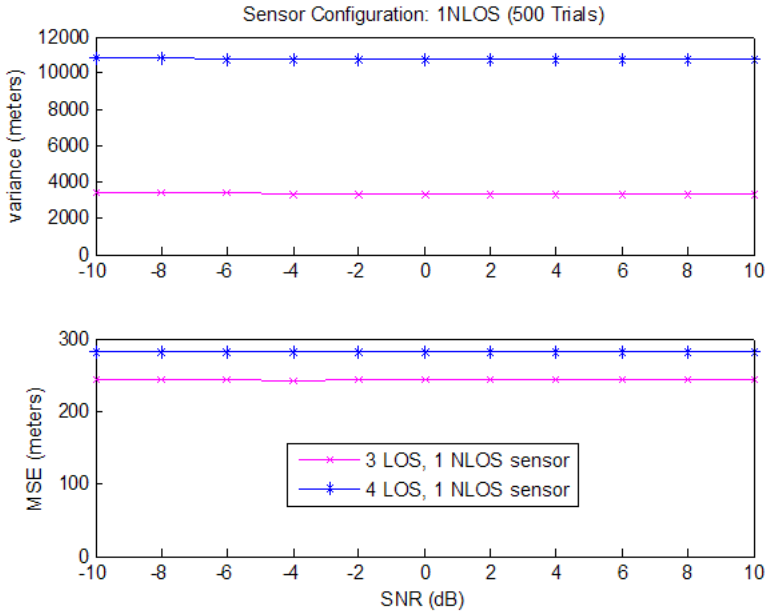


Figure 4.7: NLOS configuration data. This figure shows how large the variance and error of the estimated position are when there is a NLOS sensor included in the position estimate.

4.2.5.2 Optimized Position Estimation in NLOS Configuration. Because the variance in Figure 4.7 is above the desired threshold for both cases of NLOS sensor configurations, the geolocation system determines that there is a sensor with a NLOS component. The sensor error detection and identification algorithm is then run before the central processor estimates the final position. This algorithm calculates the residual error in each sensor's respective position estimates. These residuals are ranked, and the sensor with the largest residual error is removed from subsequent position estimate calculations. After the position is re-calculated with the remaining sensor data, the resultant position estimate variance falls below the threshold allowed, causing the estimated position error to also be much lower. This confirmed that if there is just one NLOS sensor, the sensor with the highest residual variance is the NLOS sensor. The case when the sensor with the highest residual variance is not the sensor contributing the most variance in the position estimate is explored in Section 4.2.6. The results after detecting and removing the NLOS sensor from the geolocation system are shown in Figure 4.8. These results are consistent with baseline results presented in Figure 4.5

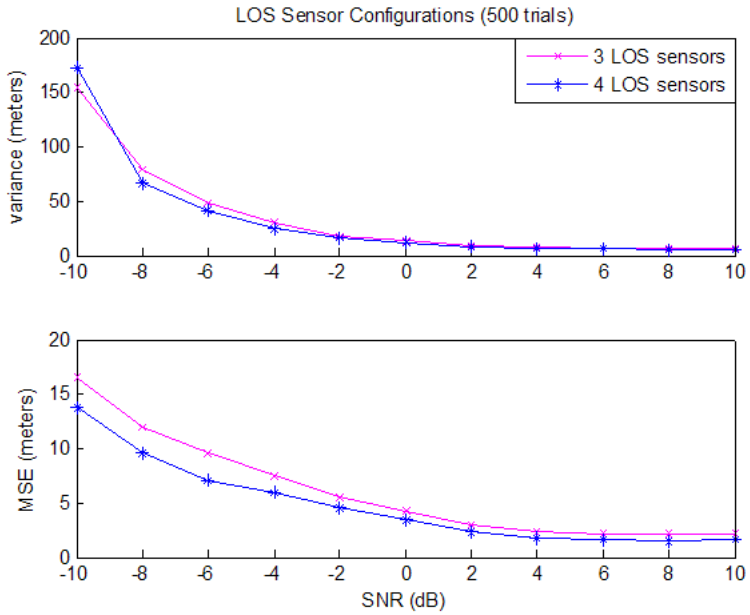


Figure 4.8: Variance and position error with NLOS sensor identified and removed from geolocation system. Remaining LOS sensors and their respective TDOAs are used to estimate the emitter's position.

4.2.6 Scenario E: Poorly Located LOS Sensor Configurations. Although the geolocation system is designed to detect and identify NLOS sensors, it can also be applied to detect the situation when a LOS sensor causes a higher variation in the position estimate, corresponding to greater error in the final estimated position. If one sensor is located further away from the emitter than all other sensors, or it is located in a bad geometrical configuration with respect to the other sensors, an increase in the variance in the position estimate may result.

Poor LOS sensor configurations, as shown in Figure 4.9, model the scenario when a group of UAV mounted sensors are approaching a suspected location of an emitter of interest. As the estimated emitter positions in Figure 4.9 show, different sensor configurations work better than others at estimating the emitter's position. All three configurations show relatively low error and variance when compared with the NLOS sensor case. Because the variance is higher than in the favorable LOS sensor scenario, it may be concluded that the sensors need to move closer to the emitter

to achieve similar variance in the position estimate and lower error in the estimated position. Figure 4.10 shows the error and variances corresponding to the poor LOS case plots.

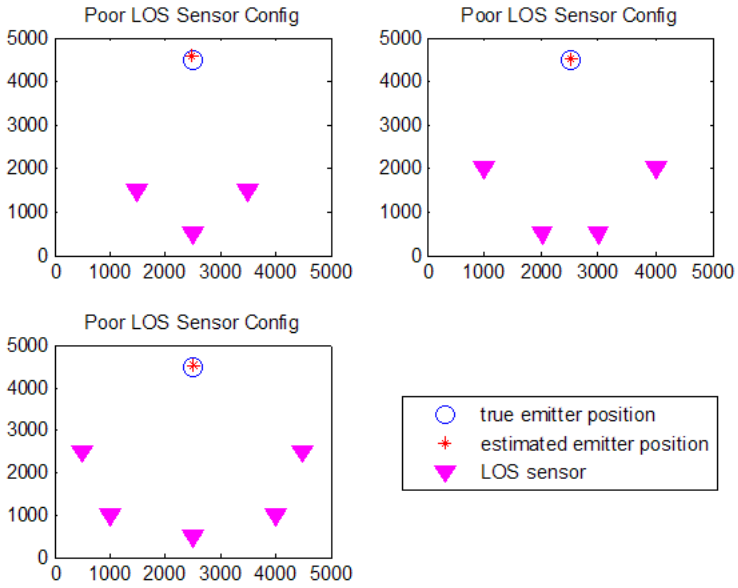


Figure 4.9: Poor LOS sensor configurations used in simulation. The estimated emitter position error is a result of all sensor TDOA information being included and used by the central processor to estimate the position of the low-power emitter ($SNR = 0\ dB$).

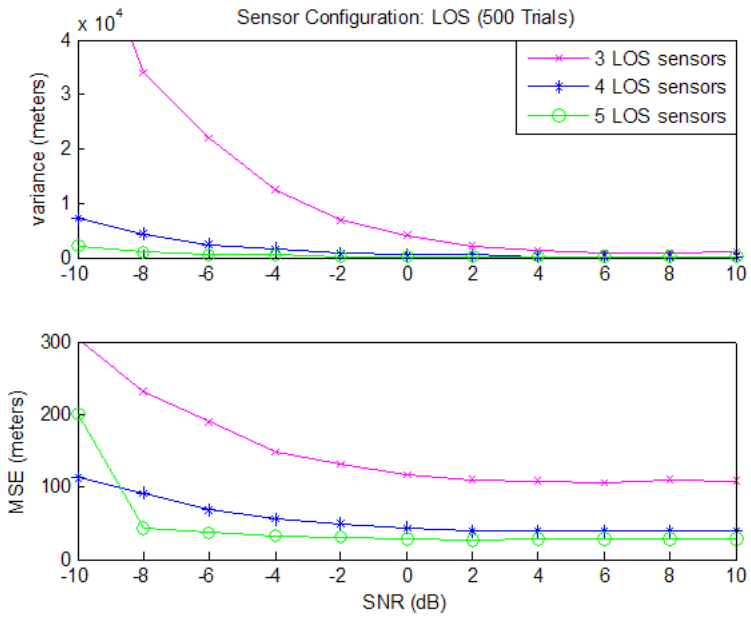


Figure 4.10: Poor LOS configuration data. This figure shows the variance and error of the estimated position when the sensors are located to one side of the emitter.

4.2.6.1 Detection and Mitigation of Poorly Located Sensors. In Figure 4.11, the resultant position error and respective variance is quite a bit larger than is observed in Figure 4.9. The sensors, located almost in a zig-zag pattern, make it more difficult for an estimator to converge to a position estimate. This is consistent with the location error ellipses developed by [8] and discussed in Chapter II.

The results in Figure 4.12 show that even when all sensors have a LOS propagation path to the emitter, there is considerable error and variance in the estimated position of the emitter. The results also show how the variance of the position estimate can indicate the existence of a poorly located LOS sensor.

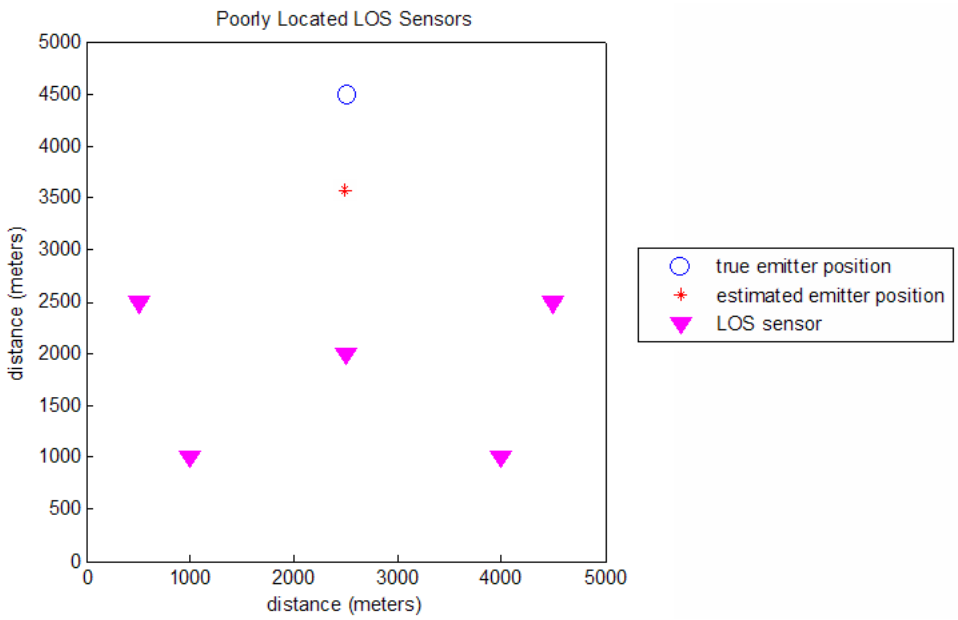


Figure 4.11: Poor LOS sensor configuration. The estimated emitter position error is a result of all sensors TDOA information being included and used by the central processor to estimate the position of the low-power emitter ($SNR = 0\ dB$).

4.2.6.2 Position Estimate with Poorly Located Sensor Removed. Because the variance of the position estimate in Figure 4.12 is observed to be above the desired variance of the favorable LOS case, the geolocation system detects that there is a poorly located LOS sensor, so the sensor error detection and identification algorithm is run before estimating the final position of the emitter. The residual error in each sensor's respective position estimates is calculated. These residuals are ranked, and the sensor with the largest residual error is removed from the geolocation system's calculations. After the position is re-calculated with the remaining sensor data, the resultant position estimate variance is compared with the variance of the favorable LOS configuration. Several sensors are removed this way, then placed back into the geolocation system before a sensor is determined to contribute the greatest variance to the position estimate. This occurred because the sensor that caused the most variance in the final position estimate does not have the highest residual. This shows that the LOS sensor having the highest residual error is not necessarily the

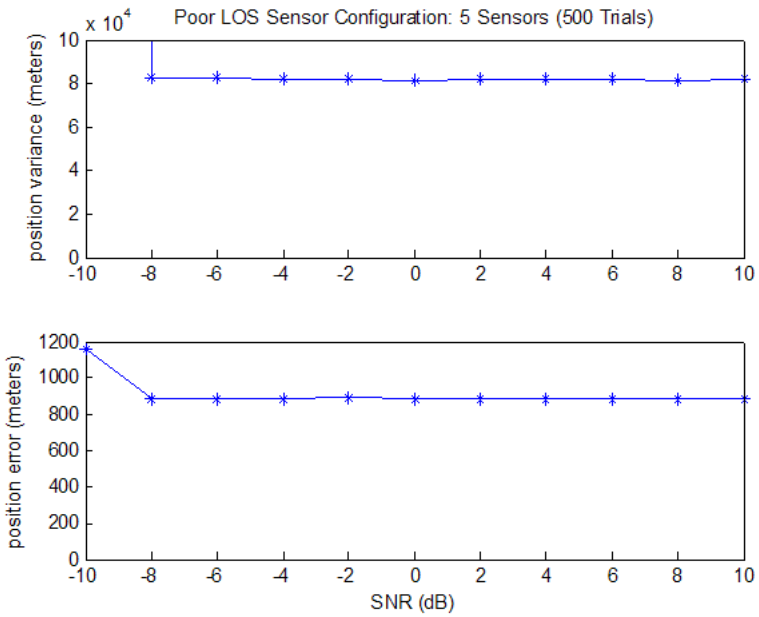


Figure 4.12: Poor zig-zag LOS (5 sensor) configuration data. This figure shows how large the variance and error of the estimated position is when the sensors are all located on one side of the emitter.

sensor that contributes most to the variance and associated error of the position estimate. The results after detecting and removing the poorly located sensor from the geolocation system are shown in Figures 4.13 and 4.14. While this position estimate is much closer to the true emitter location, it contains more than ten meters of error, and more variance associated with the position estimate. The central processor can observe this, and instruct the sensors to move to a better location that is closer to the emitter.

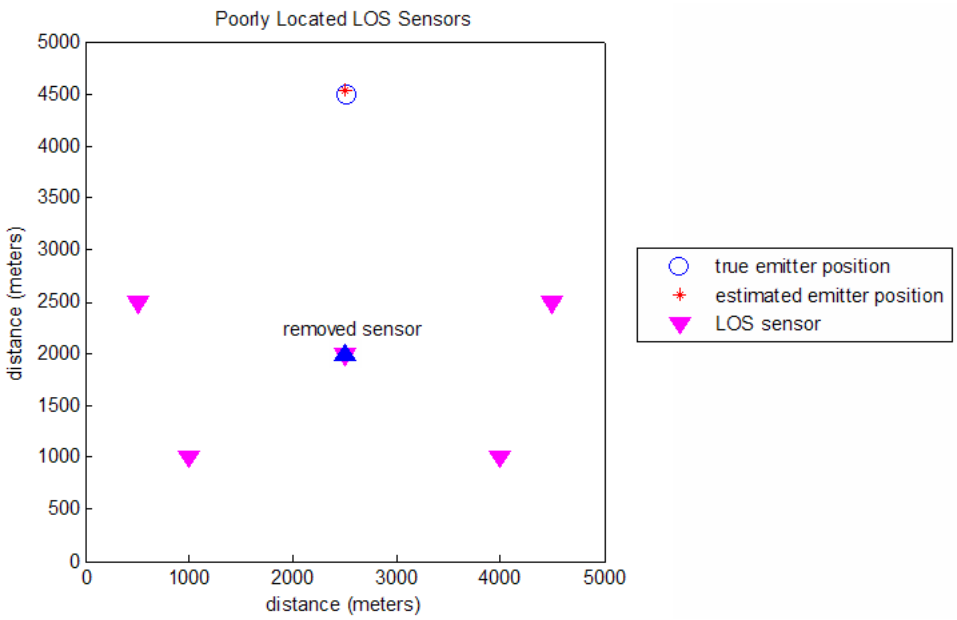


Figure 4.13: Poorly located LOS sensor removed. Estimated emitter position with poorly located sensor identified and removed from geolocation system ($SNR = 0\ dB$).

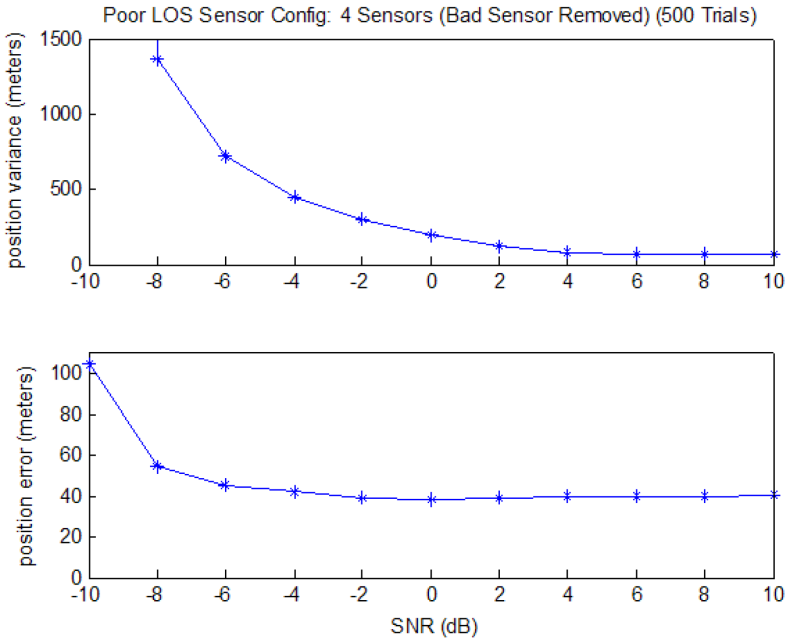


Figure 4.14: Variance and position error with poorly located LOS sensor identified and removed from zig-zag geolocation system.

4.3 Other Signal Results

Baseline results were obtained using a generated BPSK signal. While no other signal types were used to test the geolocation system, autocorrelations shown in Figure 4.15 suggest similar results can be obtained with a GSM signal.

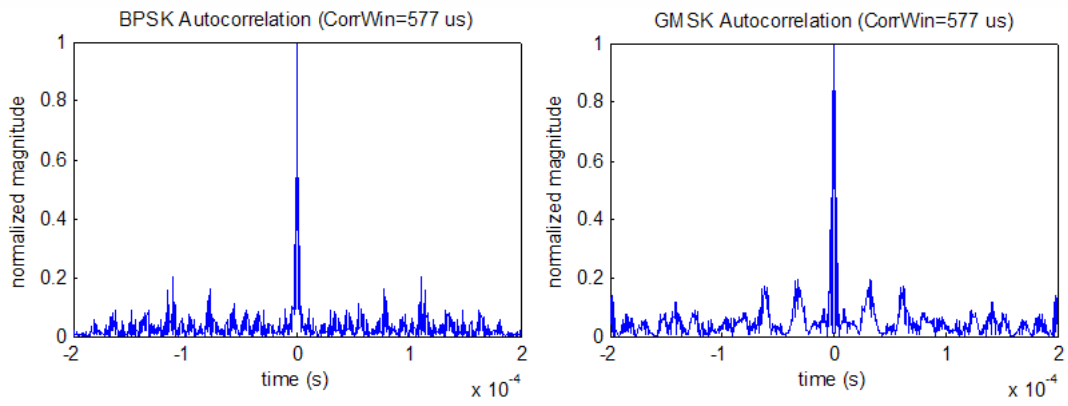


Figure 4.15: Autocorrelations of BPSK and GSM signals. These plots suggest that similar results can be optained with a GSM signal.

4.4 Summary

Based on simulation results, the ability to locate an emitter in an urban environment was demonstrated. This chapter explored the effects different sensor configurations and NLOS sensors have on the variance and accuracy of the estimated emitter position. The system was able to identify and mitigate NLOS error and poor sensor configurations using sensor residual variance.

In general, as the SNR increased the MSE of the position estimates decreased. When the SNR dropped below $-8\ dB$, significant error and variance occurred in the position estimates. By averaging across a number of Monte Carlo trials over a specified range of SNR values, a baseline containing the expected variance and position error with various number of sensors and configurations was achieved for the geolocation system.

The results presented are comparable with research completed at the Naval Post Graduate school by Mantis [17] and the ongoing work of Hippenstiel [14]. While their research goals were different, both used the TDOA method as a basis in the development of their respective geolocation systems.

Chapter V provides conclusions of this research and recommendations for future research.

V. Conclusions and Future Work Recommendations

5.1 Summary

This research presented the modeling methodology and results of a simulated passive low-power emitter geolocation system. These simulation results and analysis provide important insight that must be considered in the design and possible deployment of a geolocation system. Based on the results, a more robust geolocation system that can detect and identify the presence of NLOS signals impinging upon poorly located sensors, which add significant difficulty in passively locating emitters in the urban environment, is possible.

5.2 Conclusions

It cannot be assumed that an arbitrary emitter can be located with an arbitrary sensor configuration. If there is a very poor sensor configuration, the emitter position estimate may be very far away from the true emitter's location. In this situation, the geolocation system's estimated emitter location is incorrect with respect to range, but can provide the direction the emitter is in. This allows the system to direct the sensors closest to the emitter to obtain a better position estimate.

If the sensors are positioned around the emitter of interest, and there are at least three sensors with a LOS path from the emitter, the accuracy of the geolocation system and its ability to detect and mitigate a NLOS sensor was very good. In the case where all sensors are located on one side of the emitter, the best location estimate was found if the sensors were positioned in a semicircle facing the emitter. While not a favorable sensor configuration, this did yield more accurate position estimates than in the case when the sensors were aligned in a rough line perpendicular to the actual emitter location.

5.3 Recommendations for Future Work

While the simulated TDOA geolocation system demonstrated its ability to locate an emitter through detection and mitigation of NLOS sensors, there are still areas that could be expanded to make the system more robust. To implement the system onto a UAV or UGV, more research needs to be completed.

Several different configurations were modeled and tested with the geolocation system, but there are many more that could provide a more realistic scenario should this be implemented onto a UAV or UGV system. This will make the system more adaptable to the always changing sensor configurations.

Other types of low-power signals that are commonly used for command and control, networking, and IED detonation (garage door openers, walkie talkies, etc.) could be simulated using this geolocation system. By finding the variance and error of the resultant position estimates, it could be determined how close the sensors need to be to these types of emitters to be able to locate them with the required accuracy.

This research effort focused on the geolocation system, assuming all the communications links between the sensors and central processor were properly working. Cross-sensor communication is a key part of the system. Requirements for the central processor need to be defined, including data links, bandwidth needed, and what kind of connections will work. The determination of what geolocation tasks are assigned to whom also needs to be determined. Some of the data processing, like correlating signals, might be done at the sensors. This could minimize the amount of data they have to send to the central processor.

While a range of SNR values were used in simulating geolocation system performance, the effects of signal power loss due to propagation needs to be explored.

Listing A.1: (appendix1/GeoSystem.m)

```
%-------------------------------------------------------------
% Passive Geolocation System (GeoSystem.m)
%    Main program for passive geolocation system model.
% Written by Myrna Montminy
%-------------------------------------------------------------

clear all; clc;

%-------------------------------------------------------------
% Simulation Input Parameters

c = 3.00e008;                % speed of light (m/s)
esym = 1.0e-009;             % energy per output symbol
tsym = 3.7e-006;             % symbol duration (s)
fnot = 1/tsym;               % number of symbols (bps)
fc = 1;                      % carrier frequency (MHz)
sigamp = sqrt(2*esym/tsym);
nsamp = 100;                 % # of output waveform samples per
                             %   symbol period
fs = tsym/nsamp;             % number of samples/second

signal = 'BPSK';             % waveform used for simulation
% signal = 'GMSK';              % BPSK = baseline, GMSK = test
RxTr = 'good';               % geometric layout of sensors
%RxTr = 'baad';                  % good is surrounding emitter
                                % baad is all on one side of
                                %   emitter
numRx = 5;                   % number of sensors
ranges = [5000 5000];        % sensor configuration range
axis_lim = [0 ranges(1) 0 ranges(2)];
[Rx, Tr, reflec] = RxTrConfig(RxTr, numRx, ranges);
Rx_NLOS = [1 1 1 1 1 1];         % defines LOS (1) or NLOS (0)
SNR = [-10 -8 -6 -4 -2 0 2 4 6 8 10];
M = length(SNR);
numtrials = 500;
%-------------------------------------------------------------

%------------
for v = 1:M,    % loop running through SNR values
%------------

    %------------
    for s = 1:numtrials,    % loop running through trials
    %------------

        clear timvec; clear sigvec; clear max; clear Rx_nlos;

        %---------------------------------------------------------
        if signal == 'GMSK'     % generates GSM signal
            [timvec, sigvec] = GsmSig(1,tsym,fc,fs);
```

```
        sigvec = sigvec * sigamp;

    elseif signal == 'BPSK' % generates BPSK signal
        datsel = 'rand';    % select random data for BPSK sig
        rbits = 148;        % # of bits in one GSM sig burst
        ndelay = 0;         % # of leading samples preceeding
                            %   first valid sample
        [timvec,sigvec] = BpskSig(datsel,rbits,ndelay,...
            fnot,SNR(v),tsym,esym,nsamp,0);
    else
        signal = signal
    end

    %-------------------------------------------------------
    % sample rate information
    ts = timvec(2) - timvec(1);
    fs = 1/ts;
    CorWin = 0.577e-003/4;  % length of corr window (s)
    N = floor(CorWin/ts);   % length of corr window (# samp)
    tau = (-N+1:N-1)'*ts;

    % plot of original signal and its correlation
    % figure; hold on;
    % subplot(2,1,1);
    % plot(timvec(1:N),sigvec(1:N));
    % PlotDetails('BPSK Signal - no noise','time (s)',...
    %     'amplitude (v)',0,0,1,0);

    timvec = timvec(1:N);
    s1 = VecShift(sigvec,0);
    s1 = s1(1:N);
    sigvec(1:N) = s1;
    sigvec = sigvec';

    % subplot(2,1,2)
    % figure;
    % ww = smooth(abs(xcorr(s1,s1)),8);
    % ww = ww/max(ww);
    % plot(tau, ww);
    % axis([-2.0e-004 2.0e-004 0 1])
    % plot_details('Sampling Frequency = 2.70 MHz...
    %  (CorrWin = 144.2 us)','time (s)','magnitude',0,0,0,0);

    %-------------------------------------------------------
    % calculate true arrival time delays in received signals
    for i = 1:numRx,
        if Rx_NLOS(i) == 0,
            D(i) = ArrTime(Tr, Rx(i,:), c, reflec(i,:));
        else
            D(i) = ArrTime(Tr, Rx(i,:), c);
        end
        Ds(i) = D(i) * fs;
```

```
        sigvec_new(:,i) = VecShift(sigvec,Ds(i));
        sigvec_fnl(1:N,i) = sigvec_new(1:N,i);
    end
    sigvec = sigvec_fnl;
    clear sigvec_fnl; clear sigvec_new;

    %-------------------------------------------------------
    % generate noise
    snrat = 10^(SNR(v)/10);              % calc ratio of SNR
    sigamp = sqrt(2*esym/tsym);          % signal amplitude
    nosamp = sqrt(esym/(tsym*snrat));    % noise amplitude

    for k = 1:numRx,
        noise_new(:,k) = nosamp.*randn(size(sigvec(:,k)));
        sigvec(:,k) = sigvec(:,k) + noise_new(:,k);
    end

    %-------------------------------------------------------
    % compute cross correlation, smooth, find peak value
    TDOA_est = TdoaEst(sigvec, Rx, tau);

    %-------------------------------------------------------
    % estimate emitter position, calc residual variance wrt
    %   each receiver's position estimates
    for m = 1:numRx,
        [LocEst(m,:), LocVar(m,:), PosHis] = ...
            PosEstimator(Rx, m, Tr, c, (TDOA_est(m,:)));
        residual(m) = mean(LocVar(m,:));
        if residual(m) > 200,   % detect if sensor has NLOS
            multipath = 1;
        else
            multipath = 0;
        end
    end

    LocEstVar(s,:) = var(LocEst,1);
    PosEstVar(s) = mean(LocEstVar(s,:));

    %-------------------------------------------------------
    % take out each Rx seperately to see if estimated
    %   position converges
    if multipath > 0,
        clear LocEst;
        [Rx_multipath, RxNLOS, LocEst] = ...
            FindNlosRx(multipath,Tr,Rx,Rx_NLOS,reflec,c);
    else
        multipath = multipath;
    end

    %-------------------------------------------------------
    % final estimated location at specified SNR value
    if length(LocEstFix) > 2,
```

```
            location = mean(LocEst);
        else
            location = LocEst;
        end

        Pos(s,:) = location;
        PosErr(s) = ((Tr(1)-Pos(s,1))^2 + ...
            (Tr(2)-Pos(s,2))^2 )^(1/2);

    %------------
    end  % end of trial runs loop s
    %------------

    PosVar(v,:) = var(Pos,1);
        % variance wrt each trial position average
    PosVar(v) = mean(PosVar(v,:));
        % variance wrt each trial variance average relative to
        %   each Rx's position estimate
    PosVarRes(v) = mean(mean_var);
    PosEstVarRx(v) = mean(PosEstVar);
        % variance wrt each Rx's position estimate variance
    PosAve(v,:) = mean(Pos);
        % estimated position average over s trials
    Pos2Err(v) = mean(PosErr);
        % average estimated position error over s trials

%------------
end  % end of main process loop v
%------------

%-----------------------------------------------------------------
% plotting SNR versus position variance

close all;

figure;
subplot(211);
hold on;
plot(SNR,PosEstVarRx, '*-');
PlotDetails('Sensor Configuration: 5LOS (500 Trials)',...
    0,'position variance (meters)',0,0,0,1);
% line(xlim,[max(ylim) max(ylim)],'color','k');
% line([max(xlim) max(xlim)],ylim,'color','k');

%-----------------------------------------------------------------
% plotting SNR vs position error

subplot(212);
hold on;
plot(SNR,Pos2Err,'*-');
PlotDetails(0,'SNR (dB)',...
'position error (meters)',0,0,0,1);
```

```
    % line(xlim,[max(ylim) max(ylim)],'color','k');
    % line([max(xlim) max(xlim)],ylim,'color','k');

%-------------------------------------------------------------------
    % plotting sensors with estimated positions for each SNR value
    figure; hold on;
    %[Rx, Tr, reflec] = rx_locations(RxTr, 5, ranges);
    PlotRxTr(Rx, Tr, ranges);
%plot(Rx(3,1),Rx(3,2), 'b^', 'markersize', 10);
    for xx = 1:M,
        plot(PosAve(xx,1),PosAve(xx,2),'+');
    end
    % line(xlim,[max(ylim) max(ylim)],'color','k');
% line([max(xlim) max(xlim)],ylim,'color','k');

    % save GSM_500_LOS_3good_5wls Pos* Rx Tr;
```

Listing A.2: (appendix1/RxTrConfig.m)

```
function [Rxy, Tr, reflec] = RxTrConfig(RxTr, numRx, ranges)

%-----------------------------------------------------------------
% This function sets sensor, transmitter and reflector locations
% Written by Myrna Montminy
%
% In:     RxTr = 'good' or 'baad'
%         numRx = number of receivers
%         ranges = area receivers can be in (i.e. 5000, 5000)
%
% Out:    Rx = matrix of receiver lcoations within range
%         Tr = emitter location
%         reflec = reflector locations
%-----------------------------------------------------------------

if RxTr == 'good';
    Tr = [2500 2500];
    if numRx == 3,
        Rxy(1,:) = [2500 4500];
        Rxy(2,:) = [4500 500];
        Rxy(3,:) = [500 500];
        reflec = [3500 4000; 3500 500; 1500 500];
    elseif numRx == 4,
        Rxy(1,:) = [4000 4000];
        Rxy(2,:) = [4000 1000];
        Rxy(3,:) = [1000 1000];
        Rxy(4,:) = [1000 4000];
        reflec = [4000 3000; 4000 2000; 1000 2000; 1000 3000];
    elseif numRx == 5,
        Rxy(1,:) = [2500 4500];
        Rxy(2,:) = [4500 3000];
        Rxy(3,:) = [4000 500];
        Rxy(4,:) = [1000 500];
        Rxy(5,:) = [500 3000];
        reflec = [3500 4000; 4000 2000; 4000 2000; 1000 2000;...
            1000 2000];
    elseif numRx == 6,
        Rxy(1,:) = [3500 4500];
        Rxy(2,:) = [4500 2500];
        Rxy(3,:) = [3500 500];
        Rxy(4,:) = [1500 500];
        Rxy(5,:) = [500 2500];
        Rxy(6,:) = [1500 4500];
        reflec = [4500 2500; 3500 500; 1500 500; 500 2500;...
            1500 4500; 3500 4500];
    else
        numRx = numRx
    end

elseif RxTr == 'baad',
```

```
        Tr = [2500 4500];
        if numRx == 3,
            Rxy(1,:) = [3500 1500];
            Rxy(2,:) = [2500 500];
            Rxy(3,:) = [1500 1500];
            reflec = [3500 4000; 3500 500; 1500 500];
        elseif numRx == 4,
            Rxy(1,:) = [4000 2000];
            Rxy(2,:) = [3000 500];
            Rxy(3,:) = [2000 500];
            Rxy(4,:) = [1000 2000];
            reflec = [3500 4000; 3500 500; 1500 500];
        elseif numRx == 5,
            Rxy(1,:) = [4500 2500];
            Rxy(2,:) = [4000 1000];
            Rxy(3,:) = [2500 500];
            Rxy(4,:) = [1000 1000];
            Rxy(5,:) = [500 2500];
            reflec = [4500 4000; 4000 4000; 3500 1000; 1000 4000;...
                500 4000];
        else
            numRx = numRx
        end

    elseif RxTr == 'rand',
        for i = 1:numRx,
            Rxy(i,:) = [floor(rand*ranges(1))+trans(1); ...
                floor(rand*ranges(2))+trans(2)];
        end

        % pick Tr located within Rx ranges
        Rx_range = range(Rxy);
        Tr = [(floor(rand*Rx_range(1))+min(Rxy(:,1))) ...
            (floor(rand*Rx_range(2))+min(Rxy(:,2)))];

    else
        RxTr = RxTr
    end

    return;
```

Listing A.3: (appendix1/GsmSig.m)

```
function[GMSK_time, GMSK_sig_out, pulse1] = ...
    GsmSig(num_frames,Ts,fc,fs);

%-----------------------------------------------------------------
% This function generates a GMSK waveform signal
% Adopted from a script created by Clint Sikes
%
% In:     num_frames = 1 frame_num = 0.2354s or 51 TDMA frames
%         Ts = signal length (us)
%         fc = carrier frequency
%         fs = signal frequency
%
% Out:    GMSK_time = time vector for GSM signal
%         GMSK_sig_out = simulated GSM signal with no noise
%         pulse1 = one pulse length of signal (577 us)
%-----------------------------------------------------------------

pulselength = 577;          % Length of pulse (us)
L = 3.7;                    % Length of GMSK Pulse Shape (us)
ns = fs*Ts;                 % Number of samples/bit  (4)
BT = 0.3;                   % BT Parameter of GMSK Pulse
h = 0.5;                    % Modulation Index of GMSK Pulse
z0 = 0;                     % Initial Phase of GMSK Signal
N = floor(pulselength/L);   % Number of bits in T1
%N = 192;

%-----------------------------------------------------------------
% generate GMSK pulse shape
% Ts = 3.7;
tpulse = [-1.5*Ts:1/fs:1.5*Ts-1/fs];
g = 1/(2*Ts).*(qfunc(2*pi*BT.*(tpulse-Ts/2)./...
    (Ts*sqrt(log(2))))-qfunc(2*pi*BT.*(tpulse+Ts/2)./...
    (Ts*sqrt(log(2)))));
g = g/(2*sum(g));

%-----------------------------------------------------------------
% uncomment when outputing GSM signal > 1 pulse length
% seg = [1 2 3];
% frame_len = floor(num_frames/3);
% frames = [];
% for m = 1:frame_len,
%     frames = [frames seg];
% end

GSM_sig = [];
% for k = 1:length(frames),

    % Generate 4 pulses to simulate
    pulse4 = [];
    for j = 1:4,
```

```
            clear sGMSK;
            clear bits;
            % generate 1 pulse one time slot (TS) long (15/26 ms)
            % generate vector of binary bits
            bitsin=round(rand(1,N))';
            % Converting bits to NRZ
            for i=1:N
                if bitsin(i)==0
                    bits(i)=-1;
                else
                    bits(i)=1;
                end
            end
            bits=bits';
            % generate SOI
            [Phase,sGMSK]=GmskMod(L,bits,ns,fc,Ts,N,BT,g,h);
            pulse = sGMSK;
            % TDMA = 8 time slots
            zero_vec = zeros(1,7*length(sGMSK));
            pulse4 = [pulse4 pulse zero_vec];
%            time_p4 = time_p4 + 8;
            pulse1(j,:) = pulse;
        end
        time4 = linspace(0,0.000577*8*4,length(pulse4));

    %-----------------------------------------------------------------
    % uncomment when outputing GSM signal > 1 pulse length
    %     % implement typical spacing while phone is active
    % % %    time_len = 0;
    %     if frames(k) == 1,
    %         zero_vec1 = nosamp.*randn(1,15*length(pulse4));
    %         GSM_sig = [GSM_sig new_noisy_GMSK zero_vec1];
    % % %         time_len = time_len + (16*time_p4);
    %     elseif frames(k) == 2,
    %         zero_vec2 = nosamp.*randn(1,34*length(pulse4));
    %         GSM_sig = [GSM_sig new_noisy_GMSK zero_vec2];
    % % %         time_len = time_len + (35*time_p4);
    %     else,
    %         zero_vec3 = nosamp.*randn(1,50*length(pulse4));
    %         GSM_sig = [GSM_sig new_noisy_GMSK zero_vec3];
    % % %         time_len = time_len + (51*time_p4);
    %     end

        % implement typical spacing while phone is active
    %    time_len = 0;
    %     if frames(k) == 1,
    %         zero_vec1 = zeros(1,15*length(pulse4));
    %         GSM_sig = [GSM_sig pulse4 zero_vec1];
    % % %         time_len = time_len + (16*time_p4);
    %     elseif frames(k) == 2,
    %         zero_vec2 = zeros(1,34*length(pulse4));
    %         GSM_sig = [GSM_sig pulse4 zero_vec2];
```

```
% %         time_len = time_len + (35*time_p4);
%     else,
%         zero_vec3 = zeros(1,50*length(pulse4));
%         GSM_sig = [GSM_sig pulse4 zero_vec3];
% %         time_len = time_len + (51*time_p4);
%     end
%
%
% end

% time_len = time_len * 0.000577;
time_len = num_frames * 51 * 8 * 0.000577;
% time = linspace(0,(0.000570*4*8*51*6),length(GSM_sig));
time = linspace(0,time_len,length(GSM_sig));

% data is large - reshape to output to main file
% GMSK_sig_out = reshape(GSM_sig, length(GSM_sig)/...
%   length(pulse4), (length(pulse4)));
% GMSK_time = reshape(time, length(time)/length(pulse4), ...
%   (length(pulse4)));

GMSK_sig_out = pulse4;
GMSK_time = time4;

return;
```

Listing A.4: (appendix1/GmskMod.m)

```
function [Qt,Rt] = GmskMod(L,a,ns,fc,Ts,N,BT,g,h);

%-----------------------------------------------------------------
% This Function Generates a GMSK FH Signal
% Adopted from a Script Created by Jocelyn Dobson
%   Mar 06 - Modified by Clink Sikes
%   29 Nov 06 - Modified by Myrna Montminy
%
% In:   L = length of GMSK pulse shape in microseconds
%       a = bits = bitstring to be modulated
%       ns = number of samples / bit
%       fc = carrier frequency
%       Ts = symbol period, default is 1 at 2 Mbps
%       hoplength = hoprate = T2
%       N = number of bits in T1
%       BT = BT parameter of GMSK pulse
%       g = one GMSK pulse
%       h = modulation index of GMSK pulse
%
% Out:  Qt = phase
%       Rt = GMSK signal
%-----------------------------------------------------------------

Rt=[];
fs=ns/Ts;
rd = zeros(L-1,1);  % data vector tail
Q0 = 0;  % phase at the end of the bit

% Generate the random data
datain = [rd; a];
rd = datain(N+1 : N+L-1);

% Generate the phase shape during one period T
% Phase segmentation, corresponding to q(t-iT) for i = 3 to 1
q = cumsum(g);  % g is the Gaussian filter function
length(q)
qg = reshape(q, ns, L)';
qg = qg(L:-1:1,:);

% First term of phase equation
Qt = pi*(datain(1:N)*qg(1,:) +datain(2:N+1)*qg(2,:)+...
    datain(3:N+2)*qg(3,:));
Qt = reshape(Qt', 1, N*ns);  % arrange into 1D vector

% Generate the phase offset at the end of bit
% Second term of phase equation
S = cumsum([Q0; datain(1:N)]);
Q0 = S(N+1);                % save phase at end of last bit
S = S(1:N)'*pi/2;           % normalise by pi/2
Q1 = S(ones(1, ns),:);      % interpolation for sampling
```

```
    Q1 = Q1(:)';
    % Combine to give the final phase
    Qt = (Qt + Q1).*(h/(1/2));  %Normalize by modulation Index "h"
55
    for i=1:N
        % Form signal to be transmitted
        n = [(i-1)*Ts:1/fs:i*Ts-1/fs];  % form time base
        I = cos(2*pi*fc*n).*cos(Qt(fs*Ts*(i-1)+1:fs*Ts*i));
60          % in-phase component
        Q = sin(2*pi*fc*n).*sin(Qt(fs*Ts*(i-1)+1:fs*Ts*i));
            % quadrature component
        Rt_temp = I - Q;                % transmitted signal
        Rt=[Rt Rt_temp];
65  end

    return;
```

Listing A.5: (appendix1/BpskSig.m)

```
function [time,sigvec] = BpskSig(datsel,rbits,ndelay,fnot,...
    snr,tsym,esym,nsamp,noisamp)

%-----------------------------------------------------------------
% This function produces a sampled MPSK output waveform with
%    symbols generated per Eq 4.31 of Sklar's Digital
%    Communications text (2nd Ed.)%
%               [1] => s1 => Phz = 180 Deg
%               [0] => s2 => Phz =  0 Deg
% BPSK_MOD Function:  Binary Phase Shift Keying Modulator
%-----------------------------------------------------------------
%
% Developed: Apr 04
% Dr. Michael A. Temple
% Modified by Capt Montminy, Dec 2006
%
% In:
%       datsel - Data control variable:
%       'user' --> User SUPPLIED Data ("indata") for Modulation
%       'rand' --> Random Data GENERATED for Modulation
%       rbits - For "datsel" = 'rand' ... Number of "rbits"
%       Randomly generated for Modulation
%       ndelay - Number of LEADING samples preceeding first valid
%       sample of first Complete Symbol
%       fnot - Modulator Output Frequency (Hertz)
%       snr - Input Signal-to-Noise Ratio in Decibels (dB)
%       tsym - SYMbol Time / Duration
%       esym - Energy per putput SYMbol
%       nsamp - Number of output waveform SAMPles per Symbol
%       Period (tsym)
%   noisamp - amplitude of noise wrt signal amplitude
%
% Out:
%   time - sample time vector
%   sigvec - modulated signal vector
%-----------------------------------------------------------------

msym = 2;  % sets default to BPSK (M=2)
wnot = 2*pi*fnot;  % radian frequency of carrier
snrat = 10^(snr/10);  % calculate tatio form of input SNR
sigamp = sqrt(2*esym/tsym);  % signal component amplitude
nosamp = sqrt(esym/(snrat*tsym));  % amplitude of received
                                   % noise term

% initial SYNChronization BIT vector (syncbit)
 syncbit = [1,1,1,0,0,0,1,0,0,1,0];  % 11-Bit "Barker Sequence"

% IF datsel = "user" use "indata" vector as "rdata" vector
rdata = [];  % Initialize Data Vector
if datsel=='user'
```

```
      rdata = indata;
      rdata = cat(2,syncbit,rdata);
      rbits = length(rdata);
      rbits = rbits - mod(rbits,2);
   end

   % IF datsel = "rand" generate a Random "rdata" vector of length
   %   "rbits"
   if datsel=='rand'
      rbits = rbits - mod(rbits,2);
      for jbits = 1:rbits  % (rbits-4)
         y = rand;
         if(y<0.5)
            rdata(jbits) = 1;
         else
            rdata(jbits) = 0;
         end
      end
      rbits = length(rdata);
   end
   bitsin = rdata;  % actual BITS INto the Modulator

   % calculate number of symbol periods (nsym) in RDATA
   bitsym = 1;  % number of bits/symbol = 1 for BPSK
   nsym = round(rbits/bitsym);
   tstep = tsym/nsamp;
   modout = [];

   % Initialize "Sampled" Continuous Waveform vector (sigvec) and
     % total Time vector (time)with leading zero samples

   % Assign initial "LEADING" waveform sample and time values
   sigvec = [];
   %sigvec = zeros(1,ndelay);
   time = [];
   tlow = -tstep*ndelay;
   thgh = tstep*(nsym*nsamp-1);
   time = [tlow:tstep:thgh];
   ntime = length(time);  % total number of initial time values
   randn('state',sum(100*clock));  % Reset Random # Generator

   %-----------------------------------------------------------------
   % Calculate Sampled Output Waveform over each "Symbol Interval"
   %-----------------------------------------------------------------

   % Initialize Symbol TIMe INTerval (timint), 0 < t < Tsym
   timint = [];
   tlow = 0;
   thgh = tstep*(nsamp-1);
   timint = [tlow:tstep:thgh];
   %
   for jsym = 1:bitsym:nsym*bitsym
```

```
        % Note: jsym step size = BITSYM
    symbol = sigamp*cos(wnot*timint + pi*rdata(jsym));
        % Symbol over symbol time (timint)
    sigvec = [sigvec,symbol];
        % Create Composite "Sampled" Modulated Signal vector
end

%-----------------------------------------------------------------
% End of main "Symbol Interval" Loop Calculations
%-----------------------------------------------------------------

% Fill "LEADING" samples of "sigvec" with last symbol data
if ndelay > 0
    sigvec(1:ndelay)=sigvec((length(sigvec)-ndelay+1):...
        length(sigvec));
end
% Generate and add zero mean AWGN vector over total signal
%   interval.
% Noise amplitude (nosamp) is calculated using inputs: snr,
%   esym, tsym
if noisamp == 1,
    noise = nosamp*randn(size(sigvec));
else
    noise = 0;
end
sigvec = sigvec + noise;
% TOC % Stop Subroutine Timer
return
```

Listing A.6: (appendix1/PlotDetails.m)

```
function PlotDetails(titlestr, xlabelstr, ylabelstr, ...
    axis_lim, axis_specs, hold_specs, grid_specs)

%-------------------------------------------------------------------
% This function plots the details for a figure
% Written by Myrna Montminy
%
% In:     title = title(*)
%         xlabel = xlabel(*)
%         ylabel = ylabel(*)
%         axis_lim = axis(*)
%         axis_apecs = square(1), image(2), both(12), no specs(0)
%         hold = on(1) / off(0)
%         grid_specs = on(1) / off(0)
%
% Out:    details plotted on desired graph
%-------------------------------------------------------------------

if titlestr == 0,
    titlestr = titlestr;
else
    title(titlestr);
end
if xlabelstr == 0,
    xlabelstr = xlabelstr;
else
    xlabel(xlabelstr);
end
if ylabelstr == 0,
    ylabelstr = ylabelstr;
else
    ylabel(ylabelstr);
end
if axis_lim == 0,
    axis_lim = axis_lim;
else
    axis(axis_lim);
end
if axis_specs == 1,
    axis square;
elseif axis_specs == 2,
    axis image;
elseif axis_specs == 12,
    axis image;
    axis square;
else
    axis_specs = axis_specs;
end
if hold_specs == 0,
    hold off;
```

```
else
    hold on;
end
if grid_specs == 1,
    grid on;
else
    grid_specs = grid_specs;
end

return
```

Listing A.7: (appendix1/smooth.m)

```
function y = smooth(x,M)

%-------------------------------------------------------------------
% This function smoothes the input vector by taking the moving
%   average over the amount specified
% Written by Dr. Robert Mills, Jul 06
%
% In:   x = Input vector
%       M = smoothing factor (even)
%
% Out:  Y = smoothed version of X
%-------------------------------------------------------------------

if M < 2
  y=x;
else
  N=length(x);  % get length of input vector
  s=ones(1,M+1)/(M+1);
  y=conv(s,x);
  y=y( 1+M/2:N+M/2 );
end
return
```

Listing A.8: (appendix1/ArrTime.m)

```
function [sig_time] = ArrTime(Tr, R1, c, refl)

%-----------------------------------------------------------------
% This function calculates the true arrival time of the signal
%   at each of the sensors
% Written by Myrna Montminy
%
% In:     Tr = transmitter location
%         R1 = receiver location
%         refl = reflection location
%         c = speed of light in feet/sec
%
% Out:    sig_time = arrival time of the signal
%
%-----------------------------------------------------------------

% distance between transmiter of interest and receiver
D = (((R1(2)-Tr(2))^2)+((R1(1)-Tr(1))^2))^0.5;

% added propagation time due to a reflection
if nargin == 3,
    D = D;
else
    % added propagation time from multipath errors
    Dx = (((Tr(2)-refl(2))^2)+((Tr(1)-refl(1))^2))^0.5;
        D = Dx + (((R1(2)-refl(2))^2)+((R1(1)-refl(1))^2))^0.5;
    hold on;
    plot(refl(1),refl(2),'s');
end

% propagation time of signal from transmitter to receiver
sig_time = D / c;

return;
```

Listing A.9: (appendix1/VecShift.m)

```
function [Y] = VecShift(X,d)

%---------------------------------------------------------------
% This function shifts the input vector by the amount
%   specified.  Zeros are shifted in one end and data is lost
%   out the other (i.e., no circular shift). Input vector can be
%   row or column.
% Written by Dr. Robert Mills, Jul 06
% Modified by Myrna Montminy, Oct 06
%
% In:    X = Input vector
%        d = shift value ( >0 = shift right )
%
% Out:  Y = shifted version of X
%---------------------------------------------------------------

Xrow = X(:)';
N = length(Xrow);
N = 2*N;
ad = abs(d);
Zrow = zeros(1,ad);
Zpad = zeros(1,N);

if d == 0
        Yrow = [Xrow Zpad];
    Y = Yrow(1:N);
elseif d > 0
        Yrow = [Zrow Xrow Zpad];
        Y = Yrow(1:N);
elseif d < 0
        Yrow = [Xrow Zrow];
        Yrow = Yrow( ad+1:ad+N );
end

% return Y as either row or column depending on input
Y = Y';
return
```

Listing A.10: (appendix1/TdoaEst.m)

```
function [TDOA_est] = TdoaEst(GSM_sig, Rx, tau)

%-------------------------------------------------------------------
% Generalized Correlation Method for Estimation of Time Delay
% Knapp and Carter, IEEE Trans ASSP-24, Aug 1976
%   addapted for this research modified 11 Oct 06
%   added affects of multipath to TDOA estimates 15 Oct
%   added GSM simulated test signal
%-------------------------------------------------------------------
% In:   GSM_sig = signals to be correlated
%       Rx = reference sensor
%       tau = time vector
%
% Out:  TDOA_est = estimated TDOA between 2 signals
%-------------------------------------------------------------------

sm = 8;  % smoothing factor
numRx = length(Rx);
figure; hold on;

for i = 1:numRx,
    for j = 1:numRx-1,
        clear k; clear R; clear Rsm; clear TDOA; clear RSM_max;
        k = i+j;
        if k > (numRx),
            k = k - (numRx);
        else
            k = k;
        end

        % compute cross correlation, smooth, find peak value
        R = abs(xcorr(GSM_sig(:,i), GSM_sig(:,k)));
        Rsm = smooth(R,sm);  % smooth out
        [R_max II] = max(Rsm);
        TDOA = tau(II);  % time delay for the max value
        TDOA = -TDOA;
        TDOA_est(i,j) = max(TDOA);
    end
end

return;
```

Listing A.11: (appendix1/FindNlosRx.m)

```
function [multipath, Rx_nlos, est_pos] = ...
    FindNlosRx(multipath, Tr, Rx, Rx_NLOS, reflec, c)

%-------------------------------------------------------------------
% This function test the given sensor for NLOS by removing it
%   and re-estimating the position.  This function continues
%   to be called until a NLOS sensor is identified and removed
%   and the variance is found below desired max.
% Written by Myrna Montminy
%
% In:   multipath = 1 for 1 NLOS detected, 2 for 2 NLOS detected
%       Tr = transmitter location
%       Rx = 3 receiver locations
%       Rx_NLOS = sensor being tested for NLOS
%       reflec = reflector
%       c = speed of light (m/s)
%
% Out:  multipath = 0 for no NLOS detected
%       Rx_nlos = sensor identified as NLOS
%       est_post = estimated location of emitter
%-------------------------------------------------------------------

numRx = length(Rx);
% take out each Rx seperately to see if est pos converges
n = 1;  % Rx1 thru every Rx until multipath Rx is found
        %   if mult Rx is not found, assume 2 mult. Rx's exist

while multipath == 1,
    Rxx = circshift(Rx, ((n)*(-1)));
    Rxx = Rxx(1:(numRx-1), :);
    Rx_NLOSx = circshift(Rx_NLOS', ((n)*(-1)));
    Rx_NLOSx = Rx_NLOSx';
    Rx_NLOSx = Rx_NLOSx(1: (numRx-1));
    reflecX = circshift(reflec, (n*(-1)));
    reflecX = reflecX(1:(numRx-1), :);
    [TDOA_true, TDOA_outX] = TdoaEst(Tr, Rxx, c, ...
        reflecX, Rx_NLOSx);
    [location_est, location_est_std] = ...
        PosEstimator(Rxx, 1, Tr, c, (TDOA_outX(1,:)));
    if location_est_std(1) > 100,
        multipath = 1;
    elseif location_est_std(2) > 100,
        multipath = 1;
    else
        multipath = 0;
        Rx_multipath = n;
        Rx_nlos_xx = 0;
        est_pos(n,:) = location_est;
    end
    clear Rxx; clear Rx_NLOSx; clear reflecX; clear TDOA_outX;
```

```
        n = n+1;
        if n > (numRx+1),
            multipath = 0;  % !!  need to change this back to 2 !!
            % after each Rx is pulled to test for multipath, if
            %   there is still large std in the estimated position,
            %   then there are 2 Rx's with multipath
        else
            multipath = multipath;
        end
    end  % end of 1 NLOS test while loop

    p = 1;
    % 2 Rx's with NLOS:
    while multipath == 2,
        Rx1 = circshift(Rx, ((p)*(-1)));
        Rx1 = Rx1(1:(numRx-1), :);
        Rx_NLOSx1 = (circshift(Rx_NLOS', ((p)*(-1))))';
        Rx_NLOSx1 = Rx_NLOSx1(1: (numRx-1));
        reflecX1 = circshift(reflec, (p*(-1)));
        reflecX1 = reflecX1(1:(numRx-1), :);
        for i = 2:numRx,
            Rx2 = circshift(Rx1, ((i-2)*(-1)));
            Rx2 = Rx2(2:(numRx-1), :);
            Rx_NLOSx2 = (circshift(Rx_NLOSx1', ((i-2)*(-1))))';
            Rx_NLOSx2 = Rx_NLOSx2(2: (numRx-1));
            reflecX2 = circshift(reflecX1, ((i-2)*(-1)));
            reflecX2 = reflecX2(2:(numRx-1), :);
            [TDOA_true, TDOA_outX2] = TdoaEst(Tr, Rx2, c, ...
                reflecX2, Rx_NLOSx2);
            [location_Rx2, location_Rx2_std] = ...
                PosEstimator(Rx2, 1, Tr, c, ...
                (TDOA_outX2(1,:)));
            if location_Rx2_std(1) > 100,
                multipath = 2;
            elseif location_Rx2_std(2) > 100,
                multipath = 2;
            else
                multipath = 0;
                multipath_Rx(1,:) = p
                multipath_Rx(2,:) = i
                est_pos2(i,:) = location_Rx2;
            end
            clear Rx2; clear Rx_NLOSx2; clear reflecX2;
            clear TDOA_outX2;
        end
        clear Rx1; clear Rx_NLOSx1; clear reflecX1;
        p = p+1;
        if p > (numRx+1),
            multipath = 3;
            % after each Rx is pulled to test for multipath, if
            %   there is still large std in the estimated position,
            %   then there are 2 Rx's with multipath
```

```
        else
105         multipath = multipath;
        end
    end  % end of 2 NLOS test while loop

    return
```

Listing A.12: (appendix1/PosEstimator.m)

```
function [location_est, location_Rx_var, LocHis] = ...
    PosEstimator(Rx, Rx_num, Tr, c, TDOAs)

%-------------------------------------------------------------------
% This function uses the maximum likelihood method to estimate
%   the position from TDOA measurements.  A different location
%   is estimated from each 3-wise combonation of TDOA
%   measurments inputed.
% Written by Myrna Montminy
%
% In:   Rx = all receiver locations
%           matrix of all locations (up to 6 receiver locations)
%       Rx_num = Rx referenced for location estimation
%       Tr = emitter location (true)
%       c = speed of light
%       TDOAs = TDOA between all receiver combinations
%           vector of specified Rx with all other Rx's (in time)
%
% Out:  location_est = estimated location of emitter
%       location_Rx_std = standard deviation of estimated location
%-------------------------------------------------------------------

numRx = length(Rx);
Rxx = circshift(Rx,((Rx_num-1)*(-1)));

if numRx == 3,
    Rx1 = [Rxx(1,:); Rxx(2,:); Rxx(3,:)];  % Rx 1 2 3
    TDOA1 = [TDOAs(1); TDOAs(2)];
    [location_est, LocHis] = WlsEst(Rx1, Tr, c, TDOA1);

elseif numRx == 4,
    % estimated location referenced to specified sensor, Rx_num
    Rx1 = [Rxx(1,:); Rxx(2,:); Rxx(3,:)];  % Rx 1 2 3
    TDOA1 = [TDOAs(1); TDOAs(2)];
    [location_est(1,:), LocHis] = WlsEst(Rx1, Tr, c, TDOA1);

    Rx2 = [Rxx(1,:); Rxx(2,:); Rxx(4,:)];  % Rx 1 2 4
    TDOA2 = [TDOAs(1); TDOAs(3)];
    location_est(2,:) = WlsEst(Rx2, Tr, c, TDOA2);

    Rx3 = [Rxx(1,:); Rxx(3,:); Rxx(4,:)];  % Rx 1 3 4
    TDOA3 = [TDOAs(2); TDOAs(3)];
    location_est(3,:) = WlsEst(Rx3, Tr, c, TDOA3);

elseif numRx == 5,
    Rx1 = [Rxx(1,:); Rxx(2,:); Rxx(3,:)];  % Rx 1 2 3
    TDOA1 = [TDOAs(1); TDOAs(2)];
    [location_est(1,:), LocHis] = WlsEst(Rx1, Tr, c, TDOA1);

    Rx2 = [Rxx(1,:); Rxx(2,:); Rxx(4,:)];  % Rx 1 2 4
```

```
        TDOA2 = [TDOAs(1); TDOAs(3)];
        location_est(2,:) = WlsEst(Rx2, Tr, c, TDOA2);

        Rx3 = [Rxx(1,:); Rxx(2,:); Rxx(5,:)];  % Rx 1 2 5
        TDOA3 = [TDOAs(1); TDOAs(4)];
        location_est(3,:) = WlsEst(Rx3, Tr, c, TDOA3);

        Rx4 = [Rxx(1,:); Rxx(3,:); Rxx(4,:)];  % Rx 1 3 4
        TDOA4 = [TDOAs(2); TDOAs(3)];
        location_est(4,:) = WlsEst(Rx4, Tr, c, TDOA4);

        Rx5 = [Rxx(1,:); Rxx(3,:); Rxx(5,:)];  % Rx 1 3 5
        TDOA5 = [TDOAs(2); TDOAs(4)];
        location_est(5,:) = WlsEst(Rx5, Tr, c, TDOA5);

        Rx6 = [Rxx(1,:); Rxx(4,:); Rxx(5,:)];  % Rx 1 4 5
        TDOA6 = [TDOAs(3); TDOAs(4)];
        location_est(6,:) = WlsEst(Rx6, Tr, c, TDOA6);

    elseif numRx == 6,
        Rx1 = [Rxx(1,:); Rxx(2,:); Rxx(3,:)];  % Rx 1 2 3
        TDOA1 = [TDOAs(1); TDOAs(2)];
        [location_est(1,:), LocHis] = WlsEst(Rx1, Tr, c, TDOA1);

        Rx2 = [Rxx(1,:); Rxx(2,:); Rxx(4,:)];  % Rx 1 2 4
        TDOA2 = [TDOAs(1); TDOAs(3)];
        location_est(2,:) = WlsEst(Rx2, Tr, c, TDOA2);

        Rx3 = [Rxx(1,:); Rxx(2,:); Rxx(5,:)];  % Rx 1 2 5
        TDOA3 = [TDOAs(1); TDOAs(4)];
        location_est(3,:) = WlsEst(Rx3, Tr, c, TDOA3);

        Rx4 = [Rxx(1,:); Rxx(2,:); Rxx(6,:)];  % Rx 1 2 6
        TDOA4 = [TDOAs(1); TDOAs(5)];
        location_est(4,:) = WlsEst(Rx4, Tr, c, TDOA4);

        Rx5 = [Rxx(1,:); Rxx(3,:); Rxx(4,:)];  % Rx 1 3 4
        TDOA5 = [TDOAs(2); TDOAs(3)];
        location_est(5,:) = WlsEst(Rx5, Tr, c, TDOA5);

        Rx6 = [Rxx(1,:); Rxx(3,:); Rxx(5,:)];  % Rx 1 3 5
        TDOA6 = [TDOAs(2); TDOAs(4)];
        location_est(6,:) = WlsEst(Rx6, Tr, c, TDOA6);

        Rx7 = [Rxx(1,:); Rxx(3,:); Rxx(6,:)];  % Rx 1 3 6
        TDOA7 = [TDOAs(2); TDOAs(5)];
        location_est(7,:) = WlsEst(Rx7, Tr, c, TDOA7);

        Rx8 = [Rxx(1,:); Rxx(4,:); Rxx(5,:)];  % Rx 1 4 5
        TDOA8 = [TDOAs(3); TDOAs(4)];
        location_est(8,:) = WlsEst(Rx8, Tr, c, TDOA8);
```

```
        Rx9 = [Rxx(1,:); Rxx(4,:); Rxx(6,:)];  % Rx 1 4 6
        TDOA9 = [TDOAs(3); TDOAs(5)];
        location_est(9,:) = WlsEst(Rx9, Tr, c, TDOA9);

    else
        numRx = numRx;
    end

    location_est;

    % standard deviation (see Wyli 96)
    if length(location_est) == 2,
        location_est = location_est;
        location_Rx_var = [0 0];
    else
        location_Rx_var = var(location_est,1);
    end

    num_loc = length(location_est);
    if length(location_est) == 2,
        location_est = location_est';
    %    plot(location_est(1),location_est(2), '+');
    else
    %      for k = 1:num_loc,
    %          plot(location_est(k,1),location_est(k,2), 'r+');
    %      end
        location_est = mean(location_est);
    end

    % ranges = [5000 5000];
    % axis_lim = [0 ranges(1) 0 ranges(2)];

    % plot_details('Estimated Positions wrt each Rx','distance',...
    %     'distance',axis_lim,1,0,1);

    %plot_details(0,'distance (m)',...
    %    'distance (m)',axis_lim,1,0,1);

    % location_est = mean(location_est);
    return;
```

Listing A.13: (appendix1/WlsEst.m)

```
function [location_est, Ro_history] = ...
    WlsEst(Rx, Tr, c, TDOAs)

%--------------------------------------------------------------------
% Adopted from:
%   Statistical Theory of Passive Location Systems
%   Torrieri, IEEE Trans AES, Mar 1984
%       Hyperbolic Location Systems for 2D geolocation
% Modified by Myrna Montminy, Jan 07
%   position standard deviation limits added
%
% In:   Rx = 3 receiver locations
%       TDOAs = time between first Rx and additional Rx's
%       err = AWGN error in TDOAs
%
% Out:  location_est = estimated location of emitter
%--------------------------------------------------------------------

Rx = Rx';
numRx = length(Rx);  % number of receivers
TDOAs = [0; TDOAs];
Ro = [];
Ro1 = mean(Rx,2);  % Ro is initially set as midpoint between
                   %  all recievers
for j = 1:numRx,
    Ro = [Ro Ro1];
end

H = eye((numRx-1),numRx);  % creates matrix to multiply error
for i = 1:(numRx-1),
    H(i,(i+1)) = -1;
end

D = TDOAs * c;
    % relative arrival distances
Do = (((Rx(2,:)-Ro(2,:)).^2) + ((Rx(1,:)-Ro(1,:)).^2)) .^ 0.5;
    % Do = distance from station m to the reference point
F1 = (Ro-Rx);
% F = unit vector pointing from each Rx to reference point Ro
for k = 1:numRx,
     F(k,:) = F1(:,k)/Do(k);
end

Ro = Ro(:,1);
Ro_history  = Ro;

PosStd = 60;
v = 2;

while PosStd > 20,
```

```
        sigma = 1.0e-008;    % bpsk signal
        epsilon = sigma * randn(numRx,1);
        n = H * epsilon;
            % noise in receivers...
        Ht = H * D / c + n;
        for q = 1:numRx,
            N_ep(q,q) = epsilon(q);
        end
        N = H*N_ep*H';  % covariance matrix of measurement errors
        Nin = N^(-1);
        Ro1 = [];
        for j = 1:numRx,
            Ro1 = [Ro1 Ro];
        end
        Ro = Ro1;
        F1 = (Ro-Rx);
        Do = (((Rx(2,:)-Ro(2,:)).^2) + ...
            ((Rx(1,:)-Ro(1,:)).^2)) .^ 0.5;
            % Do = distance from station m to the reference point
        Do = Do';

        for k = 1:numRx,
            F(k,:) = F1(:,k)/Do(k);
        end
        Ro = Ro(:,1);
        Ro = Ro + c*((((F'*H'*Nin)*H*F)^(-1))*(F'*H'*Nin)*...
            (Ht-(H*Do/c)));
            % least squares estimator
        P = (c^2)*(((F'*H'*Nin)*H*F)^(-1));
            % covariance matrix of R_est
        Ro_history(:,v) = Ro;

        PosStdxx = std(Ro_history(:,(v-1):v),0,2);
        PosStdxx = mean(PosStdxx);
        if mean(PosStdxx) < 20,
            location_est = Ro;
            PosStd = 0.0;
        else
            PosStd = 60.0;
        end
        if v == 10,
             location_est = Ro;
             PosStd = 0.0;
        end

        clear epsilon; clear N; clear N_ep; clear FHN; clear Do;
        clear F1; clear F;
        v = v+1;
    end   % end of while loop

    return;
```

Listing A.14: (appendix1/PlotRxTr.m)

```
function plotRxTr = PlotRxTr(Rx,Tr,ranges)

%-------------------------------------------------------------------
% This function plots given receivers and sensors with range
%
% In:   Rx = receiver locations
%       Tr = true transmitter location
%       range = area receivers can be in (i.e. 5000, 5000)
%
% Out:  plot of Rx and true Tr location
%-------------------------------------------------------------------

numRx = length(Rx);
axis([0 ranges(1) 0 ranges(2)]);

%figure; hold on;
plot(Tr(1), Tr(2),'o', 'markersize', 10);

for i = 1:numRx,
    plot(Rx(i,1), Rx(i,2), 'mv', 'markersize', 10);
end

axis([0 ranges(1) 0 ranges(2)]);
grid on;

return;
```

List of Abbreviations

Abbreviation		Page
TDOA	Time Difference of Arrival	1
UAV	Unmanned Aerial Vehicle	1
UGV	Unmanned Ground Vehicle	1
AOA	Angle of Arrival	2
FDOA	Frequency Difference of Arrival	2
TOA	Time of Arrival	2
TDOA	Time Difference of Arrival	2
RF	Radio Frequency	3
LOB	Line of Bearing	6
LORAN	LOng RAnge Navigation	9
GPS	Global Positioning System	10
ML	Maximum Likelihood	16
LS	Linearization Solution	17
DAC	Divide and Conquer	19
WLS	Weighted Least Squares	20
NLOS	Non-Line of Site	22
MS	Mobile Station	22
BS	Base Station	22
TDMA	Time Division Multiple Access	35
MSE	Mean Square Error	44
BPSK	Binary Phase Shift Keyed	48
GMSK	Gaussian Minimum Shift Keyed	48
AWGN	Additive White Gaussian noise	51
SNR	Signal-to-Noise Ratio	51

Bibliography

1. "Google Maps". *NAVTEQ Map Data*, November 2006. URL `http://www.google.com/maps`.

2. Abel, J. S. "A Divide and Conquer Approach to Least-Squares Estimation". *IEEE Transactions on Aerospace and Electronic Systems*, 26(2):423–427, 1990.

3. Bard, J. D., F. M. Ham, and W. L. Jones. "An Algebraic Solution to the Time Difference of Arrival Equations". *Proceedings of the IEEE Southeastcon, Bringing Together Education, Science and Technology*, 313–319, 1996.

4. Chan, Y. T. and K. C. Ho. "A Simple and Efficient Estimator for Hyperbolic Location". *IEEE Transactions on Signal Processing*, 42(8):1905–1915, 1994.

5. Chen, P. C. "A NLOS Error Mitigation Algorithm in Location Estimation". *IEEE Wireless Communications and Networking Conference (WCNC)*, 316–320, 1999.

6. Cong, L. and W. Zhuang. "NLOS Error Mitigation in TDOA Mobile Location". *IEEE Global Telecommunications Conference (GLOBECOM)*, 1, 2001.

7. Cong, L. and W. Zhuang. "NLOS Error Mitigation in Mobile Location". *IEEE Transactions on Wireless Communications*, 4(2):560–573, 2005.

8. Drake, S., K. Brown, J. Fazackerley, and A. Finn. "Autonomous Control of Multiple UAVs for the Passive Location of Radars". *Proceedings of the 2005 International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 403–409, 2005.

9. Drake, S. R. and K. Doğançay. "Geolocation by Time Difference of Arrival Using Hyperbolic Asymptotes". *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2, 2004.

10. Fang, B. T. "Simple Solutions for Hyperbolic and Related Position Fixes". *IEEE Transactions on Aerospace and Electronic Systems*, 26(5):748–753, 1990.

11. Ferguson, B. G. "Time-Delay Estimation Techniques Applied to the Acoustic Detection of Jet Aircraft Transits". *The Journal of the Acoustical Society of America*, 106:255, 1999.

12. Foy, W. H. "Position-Location Solutions by Taylor-Series Estimation". *IEEE Transactions on Aerospace and Electronic Systems*, 12:187–194, 1976.

13. Hartwell, G. D. *Improved Geo-Spatial Resolution using a Modified Approach to the CAF*. Thesis, Naval Postgraduate School, Monterey CA, September 2005.

14. Hippenstiel, R. D., T. T. Ha, and U. Aktas. *Localization of Wireless Emitters Based on the TDOA and Wavelet Denoising*. Report, Naval Postgraduate School, Monterey CA, May 1999.

15. Knapp, C. and G. Carter. "The Generalized Correlation Method for Estimation of Time Delay". *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 24(4):320–327, 1976.

16. Krizman, K. J., T. E. Biedka, and T. S. Rappaport. "Wireless Position Location: Fundamentals, Implementation Strategies, and Sources of Error". *IEEE 47th Vehicular Technology Conference*, 2, 1997.

17. Mantis, S. D. *Localization of Wireless Communication Emitters using TDOA Methods in Noisy Channels*. Thesis, Naval Postgraduate School, Monterey CA, June 2001.

18. Marino, J. E. *Detection and characterization of Commercial GSM Intracellular Radio Frequency "Observables"*. Thesis, Air Force Institute of Technology, Wright-Patterson AFB OH, March 2006.

19. Nordlund, P. J., F. Gunnarsson, and F. Gustafsson. "Particle Filters for Positioning in Wireless Networks". *Proceedings of EUSIPCO*, September 2002.

20. Pages-Zamora, A., J. Vidal, and D. H. Brooks. "Closed-Form Solution for Positioning Based on Angle of Arrival Measurements". *The 13th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 4:1522–1526, 2002.

21. Rappaport, T. S. *Wireless Communications: Principles and Practice*. Prentice Hall, Upper Saddle River, NJ, 1996.

22. Rappaport, T. S., J. H. Reed, and B. D. Woerner. "Position Location Using Wireless Communications on Highways of the Future". *IEEE Communications Magazine*, 34(10):33–41, 1996.

23. Sikes, C. R. *Non-Cooperative Detection of Frequency-Hopped GMSK Signals*. Thesis, Air Force Institute of Technology, Wright-Patterson AFB OH, March 2006.

24. Stein, S. "Algorithms for Ambiguity Function Processing". *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(3):588–599, 1981.

25. Streight, D. A. *Application of Cyclostationary Signal Selectivity to the Carry-On Multi-Platform GPS Assisted TDOA System*. Thesis, Naval Postgraduate School, Monterey CA, March 1997.

26. Streight, D. A. *Maximum-Likelihood Estimators for the Time and Frequency Differences of Arrival of Cyclostationary Digital Communications Signals*. Dr dissertation, Naval Postgraduate School, Monterey CA, June 1999.

27. Sun, G., J. Chen, W. Guo, and K. Liu. "Signal Processing Techniques in Network-Aided Positioning". *IEEE Signal Processing Magazine*, 22(4):12–23, 2005.

28. Torrieri, D. J. "Statistical Theory of Passive Location Systems". *IEEE Transactions on Aerospace and Electronic Systems*, 20(2):183–198, 1984.

29. Urruela, A. and J. Riba. "Novel closed-form ML position estimator for hyperbolic location". *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2, 2004.

30. Vossiek, M., L. Wiebking, P. Gulden, J. Wieghardt, C. Hoffmann, P. Heide, S. Technol, and G. Munich. "Wireless Local Positioning". *IEEE Microwave Magazine*, 4(4):77–86, 2003.

31. Wu, S. Q., H. C. So, and P. C. Ching. "Improvement of TDOA Measurement using Wavelet Denoising with a Novel Thresholding Technique". *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1:539–542, 1997.

32. Wylie, M. P. and J. Holtzman. "The NLOS Problem in Mobile Location Estimation". *IEEE International Conference on Universal Personal Communications*, 2:827–831, 1996.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704–0188

1. REPORT DATE *(DD–MM–YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From — To)*
22–03–2007	Master's Thesis	Sept 2006 — Mar 2007

4. TITLE AND SUBTITLE

Passive Geolocation of Low-Power Emitters in Urban Environments Using TDOA

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Myrna B. Montminy, Capt, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way, Bldg 641
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/07-16

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/SNRW
Attn: Mr. Vasu Chakravarthy
Air Force Research Laboratory, 2241 Avionics Circle, Bldg 620
Wright-Patterson AFB, OH 45433-7321
DSN: 785-5579 x3547; e-mail: Vasu.Chakravarthy@wpafb.af.mil

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approval for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT Low-power devices are commonly used by the enemy to control Improvised Explosive Devices (IEDs), and as communications nodes for command and control. Quickly locating the source of these signals is difficult, especially in an urban environment where buildings and towers can cause interference. This research presents a geolocation system that combines several geolocation and error mitigation methods to locate an emitter in an urban environment. The proposed geolocation system uses a Time Difference of Arrival (TDOA) technique to estimate the location of the emitter of interest. Using sensors at known locations, TDOA estimates are obtained by cross-correlating the signal received at all the sensors. A Weighted Least Squares (WLS) solution is used to estimate the emitter's location. If the variance of the location estimate is too high, a sensor is detected as having a Non-Line of Sight (NLOS) path from the emitter, and is removed from the geolocation system and a new position estimate is calculated with the remaining sensor TDOA information. The performance of the system is assessed through modeling and simulations. The test results confirm the feasibility of identifying a NLOS sensor, optimizing the geolocation system's accuracy in an urban environment.

15. SUBJECT TERMS

TDOA, WLS, LOS, NLOS, multipath, passive geolocation, low-power emitter

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Dr. Robert F. Mills (ENG)
U	U	U	UU	116	19b. TELEPHONE NUMBER *(include area code)* (937)255-3636x4527,email:Robert.Mills@afit.edu